

Article

Crystallization-Inspired Design and Modeling of Self-Assembly Lattice-Formation Swarm Robotics

Zebang Pan ¹, Guilin Wen ^{2,*}, Hanfeng Yin ¹, Shan Yin ¹ and Zhao Tan ¹

¹ State Key Laboratory of Advanced Design and Manufacture for Vehicle Body, Hunan University, Changsha 410082, China; zbpan@hnu.edu.cn (Z.P.)

² School of Mechanical Engineering, Yanshan University, Qinhuangdao 066004, China

* Correspondence: glwen@ysu.edu.cn

Abstract: Self-assembly formation is a key research topic for realizing practical applications in swarm robotics. Due to its inherent complexity, designing high-performance self-assembly formation strategies and proposing corresponding macroscopic models remain formidable challenges and present an open research frontier. Taking inspiration from crystallization, this paper introduces a distributed self-assembly formation strategy by defining *free*, *moving*, *growing*, and *solid* states for robots. Robots in these states can spontaneously organize into user-specified two-dimensional shape formations with lattice structures through local interactions and communications. To address the challenges posed by complex spatial structures in modeling a macroscopic model, this work introduces the structural features estimation method. Subsequently, a corresponding non-spatial macroscopic model is developed to predict and analyze the self-assembly behavior, employing the proposed estimation method and a stock and flow diagram. Real-robot experiments and simulations validate the flexibility, scalability, and high efficiency of the proposed self-assembly formation strategy. Moreover, extensive experimental and simulation results demonstrate the model's accuracy in predicting the self-assembly process under different conditions. Model-based analysis indicates that the proposed self-assembly formation strategy can fully utilize the performance of individual robots and exhibits strong self-stability.



Citation: Pan, Z.; Wen, G.; Yin, H.; Yin, S.; Tan, Z. Crystallization-Inspired Design and Modeling of Self-Assembly Lattice-Formation Swarm Robotics. *Sensors* **2024**, *24*, 3081. <https://doi.org/10.3390/s24103081>

Academic Editor: Carlos Tavares Calafate

Received: 1 April 2024

Revised: 9 May 2024

Accepted: 10 May 2024

Published: 12 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: swarm robotics; self-assembly formation; crystallization; macroscopic model

1. Introduction

The initial inspiration for swarm robotics derives from the incredible self-organizing swarms observed in nature, such as bird flocks [1], fish schools [2], and crystallization [3]. Such natural swarms formed by numerous simple units are completely distributed and decentralized, achieving complex collective behavior solely through simple local interactions rather than external or central controls [4]. This spontaneous phenomenon is defined as self-organization, and its characteristics, including scalability, flexibility, and robustness, serve as critical indicators for developing swarm robotics [5–8]. As a result, the focal point of the field of swarm robotics is to develop a swarm of simple robots. They can go beyond the capabilities of individual robots and effectively collaborate to achieve higher-level collective objectives [6]. Over the past two decades, swarm robotics has gradually evolved into a mature field, engaging researchers worldwide in its development. A comprehensive exploration of fundamental swarm behaviors, such as aggregation [9], foraging [10], and collective exploration [11], has been undertaken. The characteristics of these swarm behaviors are also analyzed by modeling microscopic [12,13] and macroscopic [14–16] models. Through these investigations, a more profound comprehension of swarm robotics and its prospective applications has been achieved.

Self-assembly is a reversible process in which pre-existing simple entities spontaneously form an ordered spatial structure without external intervention [17,18]. In the field

of swarm robotics, self-assembly formation is considered a fundamental research topic and holds significant potential for practical applications [19]. These applications encompass the self-assembly control of the satellites [20,21], the programmable self-assembly matter [22,23], and advanced manufacturing rooted in self-assembly [24]. Consequently, this challenging and meaningful problem has attracted the attention of researchers worldwide striving to develop high-performance self-assembly swarm robotics. In 2014, Rubenstein et al. designed distributed local interaction rules for swarm robotics based on the finite-state machine (FSM) and achieved self-assembly formations by organizing a motion chain [25]. However, this approach operated in a single-threaded mode, resulting in extremely low efficiency. Following this, Yang et al. proposed a self-assembly formation based on a distributed algorithm by implementing two parallel motion chains [26,27]. Although this policy improved the efficiency, the high parallel potential of swarm robotics was not fully exploited. Additionally, Divband Soorati et al. designed swarm robotics to achieve the self-assembly of a tree formation [28]. Since father nodes constrained the growth of child nodes, the efficiency of the self-assembly was also hindered. In 2019, Zhu et al. defined four rules of collective behavior to form a square formation [29]. While this algorithm showed high parallelism and efficiency, its robustness was limited by the global leader. Zheng et al. [30] and Deshmukh et al. [31] discussed the self-assembly formation based on the density-feedback method, showing great scalability and high parallelism. However, the density-feedback laws relied on off-line global pattern planning [30] or a centralized controller [31], which conflicted with the flexibility and robustness of swarm robotics. The graph-based method was applied in a small-scale self-assembly by Klavins [32] and Mong-ying et al. [33]. Nevertheless, this control policy might face the challenge of dimensional explosion when applied to large-scale swarm robotics. Similarly, automated design methods were typically limited to small-scale self-assembly [34,35]. The potential field approach [36–38] might achieve distributed and high-efficient self-assembly formation. As the potential field only constrained robots to a specific area rather than certain positions, forming a formation with lattice structures was challenging. The above research on self-assembly of swarm robotics is summarized in Table 1. While the above research is valuable and offers essential guidance for exploring self-assembly formations, the potential of swarm robotics has yet to be realized. Indeed, crystallization is an exciting self-assembly phenomenon in nature, where the free particles efficiently and spontaneously form crystals with a periodic arrangement of atoms [39–41]. This phenomenon may contain the key principles for designing high-performance self-assembly formation strategies.

Table 1. An overview of research on self-assembly of swarm robotics.

Reference	Authors	Research Achievements	Methods	Limitations
Ref. [25]	Rubenstein et al.	Swarm robotics organized a single motion chain to achieve self-assembly formation.	The FSM method	Extremely low efficiency resulted from only single-threaded mode.
Refs. [26,27]	Yang et al.	Swarm robotics organized two parallel motion chains to achieve self-assembly formation.	The FSM method	Low efficiency resulted from only double-threaded mode.
Ref. [28]	Divband Soorati et al.	Swarm robotics self-assembled to the tree-like formation to search for bright areas.	The FSM method	The constraints between father and child nodes limit efficiency.

Table 1. Cont.

Reference	Authors	Research Achievements	Methods	Limitations
Ref. [29]	Zhu et al.	Swarm robotics self-assembled to a square formation.	The FSM method	The robustness is limited due to designing a global leader for swarm.
Ref. [30]	Zheng et al.	Swarm robotics self-assembled to the specified formation.	The density-feedback method	The flexibility is limited due to rely on the off-line global pattern planning.
Ref. [31]	Deshmukh et al.	Swarm robotics self-assembled to the specified 2D formation.	The density-feedback method	The robustness is limited due to rely on a centralized controller.
Ref. [32]	Klavins et al.	Swarm robotics achieved small-scale self-assembly formation.	The graph-based method	Not suitable for large scale self-assembly formation.
Ref. [33]	Mong-ying et al.	Swarm robotics achieved small-scale self-assembly formation.	The graph-based method	Not suitable for large scale self-assembly formation.
Ref. [34]	Groß et al.	Swarm robotics self-assembled into a small formation for collective transport.	The evolutionary algorithms	Not suitable for large scale self-assembly.
Ref. [35]	Sperati et al.	Swarm robotics self-assembled into a small-scale path.	The evolutionary algorithms	Not suitable for large scale self-assembly.
Ref. [36]	Khalidi et al.	Swarm robotics self-assembled to the specified path.	The potential field methods	Unable to construct a formation with lattice structures.
Ref. [37]	Cheah et al.	Swarm robotics self-assembled to the specified 2D formation.	The potential field methods	Unable to construct a formation with lattice structures.

The macroscopic model provides a crucial theoretical basis for studying swarm robotics [4]. It can directly capture the critical features of swarm behavior and predict the long-term behaviors of swarm robotics [42,43]. Hence, macroscopic models facilitate the analysis of the underlying mechanisms driving swarm behavior [42,43]. These analysis results can be further used to optimize the design of the swarm robotics and improve the robot controllers. Generally speaking, macroscopic models can be subdivided into non-spatial models and spatial models [4,43]. The non-spatial model is widely used, relying on the assumption of spatial uniformity [44]. Based on the probabilistic finite-state machine (PFSM), Martinoli et al. [14] and Konur et al. [45] presented the non-spatial model for the collaboration-based stick-pulling swarm. The steady-state conditions and optimal collaboration rate were further discussed. Schmickl et al. [46] used the non-spatial model based on the stock and flow diagram to analyze the feedback loops of the aggregation behaviors. Liu et al. proposed a non-spatial model for foraging behaviors using PFSM [16]. On this basis, Song et al. found the optimal decision rules of the foraging behaviors [10]. The above research demonstrates that non-spatial models are usually very adept at analyzing the driving mechanism of swarm behavior besides the predicted ability. The spatial model is another significant macroscopic approach for studying swarm robotics, being mathematically grounded in the Fokker–Planck or diffusion-reaction equations [7,47,48]. Since the spatial model characterizes the ensemble of trajectories for a swarm within a specific area and time, its predictions can be visualized to directly depict the anticipated shape

of the swarm robotics [4,7]. Schmickl et al. [46] and Prorok et al. [49] discussed spatial models in aggregation and coverage behaviors, respectively. They demonstrated little difference between spatial and non-spatial models in long-term predictions. Although the predictions of the non-spatial models are more visually intuitive, it is difficult to identify the driving mechanisms of swarm behaviors [46]. Meanwhile, due to the difficulty in obtaining analytical solutions, the applications of spatial models may be constrained [7].

Although macroscopic models have been applied extensively to study various swarm tasks and have shown great significance, modeling a macroscopic model for self-assembly formation remains a relatively underexplored area. Whether it is a spatial or non-spatial model, a key aspect of modeling is constructing the state transition functions (or state transition probabilities) between states [14,46]. To the authors' knowledge, two widely used methods for constructing the state transition function are the data statistics method [50–53] and the geometrical estimation method [14,16]. However, in all current studies on modeling macroscopic models for self-assembly formation, the state transition functions are often reduced to experiment-based free parameters [14] using the data statistics method [50–53]. Since the free parameters fail to explain the inner mechanisms of the state transitions, the scientific value of the macroscopic model becomes limited. The state transition function, based on the geometrical estimation method, holds the potential to elucidate its transition mechanisms. However, this method is only suitable for situations where state transitions are only influenced by the geometric description of the robot's sensing area and the density of the robots, e.g., stick-pulling [14], aggregation [46], and foraging [16]. In self-assembly formation, the state transition is also affected by the spatial structural features of the formation besides the robot's sensing area. These complex spatial structures introduce strong nonlinearity to the state transitions. Therefore, the geometrical estimation method cannot be used to construct state transition functions for the self-assembly formation.

Designing a high-performance self-assembly formation strategy to achieve the two-dimensional (2D) shape formation with lattice structures and presenting a corresponding macroscopic model is challenging and meaningful for advancing swarm robotics. In this paper, a high-performance distributed self-assembly formation strategy is introduced, drawing inspiration from crystallization processes. To convert the macroscopic swarm behaviors to multiple simple collaborative tasks, five states, i.e., *free*, *moving*, *building*, *growing*, and *solid*, are defined for robots inspired by the crystallization's phase transition processes (see Figure 1a and Table 2). Corresponding behavioral rules are designed for each robot state. Additionally, the concepts of unit cells and nucleation in crystallization are incorporated into self-assembly formation to initiate the self-assembly process and facilitate lattice formation, respectively. Consequently, robots in various states can autonomously collaborate and communicate locally, allowing them to form user-specified 2D shape formations with lattice structures. Notice that designing a self-assembly formation strategy for swarm robotics is not a thorough reproduction of the crystallization process due to the difference between robots and crystal. To implement the proposed self-assembly formation, Waxberry robots running within a grid ground and the embodied simulator are developed. Meanwhile, a non-spatial macroscopic model is proposed for the self-assembly formation. Here, the master equations of this model are created based on the stock and flow diagram. To solve the limitations of the geometric estimation and data statistics methods, the structural features estimation (SFE) method is proposed to construct the model's state transition functions. Through defining the basic six structural types and analyzing their features, the state transition functions that have the potential to explain the transition mechanisms are constructed. Real-robot experiments and embodied simulations verify the proposed model's predictive capability for self-assembly formation across various shapes, scales, and system parameters.

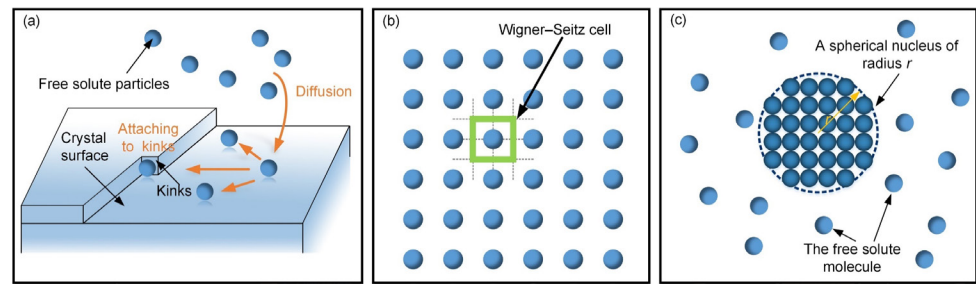


Figure 1. Three critical characteristics in crystallization. (a) The phase transition of the crystallization. Inspired by the phase transitions, the complex self-assembly formation task can be decomposed into several simple cooperative tasks among robots in different states. (b) The Wigner–Seitz unit cell in the parallelogram lattices. The unit cell is introduced for self-assembly to achieve the formation with lattice structures. (c) Nucleation. Nucleation in crystallization suggests that self-assembly formations also require a special robot to trigger this process.

Table 2. The definition of states of robot inspired by crystal growth.

The Phenomena of Crystal Growth	The States of Robot
The free solute particles	<i>Free robots</i>
The solute particles in the diffusion process	<i>Moving robots</i>
The solute particles entering kinks	<i>Building robots</i>
The atoms making up the crystal surface	<i>Growing robots</i>
The atoms inside crystal	<i>Solid robots</i>

The main contributions of this paper are summarized in the following three aspects. Firstly, a high-performance distributed self-assembly formation strategy inspired by crystallizations is proposed for homogeneous swarm robotics. Since the design of a swarm-robotics cooperative mechanism in self-assembly formation is inspired by crystallization, swarm robotics shows excellent flexibility, scalability, and high efficiency. These advantages are demonstrated by real-robot experiments and simulations. Secondly, the SFE method is proposed to address the challenges posed by complex spatial structures in modeling macroscopic models. Compared with the data statistics method, the SFE method possesses the potential to delve into the mechanisms underlying swarm behaviors. Meanwhile, the proposed method overcomes the limitations of the geometric estimation method in scenarios with spatial structures. Besides the self-assembly formation, the proposed method is also significant in modeling other swarm tasks involving complex spatial structures. Thirdly, a non-spatial macroscopic model is modeled for the proposed self-assembled formation based on the stock and flow diagram and SFE method. Compared to simulations, the proposed model can quickly and accurately predict the whole macroscopic dynamic results. This predictive capability is an important guarantee for the realization of practical applications of self-assembly formation, especially in large scale tasks. Furthermore, the analysis based on the model indicates that shorter building time consumed by a single robot means a higher forming efficiency. It points out ways of optimizing the efficiency of the self-assembly formation.

The rest of this paper is organized as follows. In Section 2, the distributed self-assembly formation strategy inspired by crystallization is proposed and verified. In Section 3, the mathematical description of the macroscopic model is introduced. In Section 4, the validation of the proposed model and discussions are expounded. Lastly, the conclusions of this paper are drawn in Section 5.

2. The Self-Assembly Formation Inspired by Crystallization

2.1. Crystallization and Inspiration

Crystallization is a natural self-assembly process in which free solute particles (e.g., atoms, ions, or molecules) are arranged into highly ordered structures, resulting in a crystal

with a periodic arrangement of atoms [39–41]. Driven by the laws of physics, free solutes spontaneously assemble into a crystal through local interactions without external controls. This phenomenon signifies a set of high-performance self-assembly rules. Consequently, it becomes feasible to devise a high-performance distributed self-assembly formation strategy for swarm robotics to create a 2D shape with lattice structures by emulating the crystallization process. Here, three critical characteristics in crystallization are worth noting, offering inspiration for self-assembly formation.

Firstly, crystallization is a phase transition [54]. Through diffusion (i.e., mass transfer), the free solute particles are transported from the solution to the growing crystal surface (see Figure 1a). According to the Kossel–Stranski model [55], the particles attached to the kinks are incorporated into the crystal by forming new bonds with crystal atoms on the surface, resulting in crystal growth. Indeed, the crystallization process can be regarded as cooperating task finished by the particles in different phases or states. This phenomenon suggests to us that the complex self-assembly formation can be decomposed into multiple simple collaborative tasks performed by robots in different states.

Secondly, the unit cell is the fundamental building block of the crystal and defines atoms' basic arrangement and repetition [56]. It is the key to generating periodic latticed structures. Here, Figure 1b shows the definition of the unit cell according to Wigner–Seitz in the parallelogram lattices [56]. Therefore, introducing the unit cell for self-assembly is the key to achieving the formation with lattice structures. Under the above inspirations of the crystallization's characteristics, the self-assembly formation strategy is introduced in Section 2.2.

Thirdly, crystallization begins in nucleation. Nucleation may occur spontaneously from the solvated phase or be artificially induced [57]. When the radius of nucleus r exceeds its critical radius, the crystal growth begins based on this stable nucleus (see Figure 1c). It is not hard to find that a stable point is essential for self-assembly.

The above characteristics can provide inspiration for designing the self-assembly formation of swarm robotics. However, due to the difference between crystals and robots, many unique crystallized mechanisms are not applied in the self-assembly formation of swarm robotics, such as the equilibrating crystallization process, the driving force of crystallization based on thermodynamics, and so on. Thus, designing the self-assembly formation strategy for swarm robotics is not a thorough reproduction and imitation of the crystallization process.

2.2. The Distributed Self-Assembly Formation Strategy for Swarm Robotics

For the self-assembly formation considered in this paper, all robots are initially distributed on the ground randomly and uniformly. This paper sets the density of robots on the ground as 0.2 by default. Generally, the number of robots is more than sufficient for self-assembly formation. Each robot has a low ability; they only know their position and can communicate with neighboring robots in a small local area. Note that direct communication has the advantages of portability and has a lower cost than stigmergy [58]. After giving the user-specified shape, robots will assemble into this formation by interacting with neighbors without relying on external control and global information. The details are designed as follows.

As introduced in Section 2.1, the inspiration drawn from crystal growth provides a framework for designing self-assembly formation. To decompose the complex self-assembly formation task into several simple cooperative tasks, five states are defined for robots inspired by the crystallization's phase transition (see Table 2). Inspired by the free solute particles, the *free* robot is designed to provide the available materials for self-assembly formation. Like solute molecules in the diffusion process, the *moving* robot is responsible for transporting itself to the growth boundary of the formation. The particles attached to the kinks inspire the design of the *building* robot, which occupies a specified position to enter and expand the formation. The *growing* and *solid* robots correspond to atoms on the surface of and inside the crystal, respectively. Considering the differences between

crystallization and self-assembly formation, the *growing* robot should provide the specified positions and recruit *free* and *moving* robots to these positions, promoting the expansion of formation. The *solid* robots no longer do anything, like the atoms inside the crystal. In addition, more sub-states are also defined for robots based on different criteria, out of consideration for modeling the macroscopic model. See Figure 2 and Section 3 for more details. As shown in Figure 1b, based on the Wigner–Seitz unit cell in the parallelogram lattices, the cross-arranged unit cells are designed for self-assembly formation to achieve the shape with lattice structures. Here, the unit cell is defined as a square grid with two unit lengths, allowing a *building*, *growing*, or *solid* robot to be positioned at its center (see Figure 3). The cross-arranged unit cells stipulate the arrangement of the robots in the formation and provide a basis for the *growing* robot to specify where it needs to be built. Referring to the definition of the cross-arranged unit cells (see Figure 3), each *growing* robot positioned in a unit cell has four neighboring unit cells. Within this context, the *growing* state is subdivided into three distinct sub-states, i.e., *G1*, *G2*, and *G3*, based on the conditions of neighboring unit cells (see Figure 2). As shown in Figure 4, the *G1*, *G2*, and *G3* states indicate that this *growing* robot has one, two, and three neighboring filled unit cells, respectively. The unit cell filled by *growing* or *solid* robots is defined as the filled unit cell; otherwise, it is the empty unit cell. These sub-states of the *growing* state are necessary for modeling the macroscopic model of self-assembly formation. For more details, see Section 3. In addition, inspired by the nucleation in crystallization, two robots will be selected as the initial *growing* robots and positioned at the coordinates (x_0, y_0) and $(x_0, y_0 + 2)$ according to the cross-arranged unit cell's definition (see Figure 5). x_0 and y_0 can be assigned any value. The configuration of two *growing* robots can trigger self-assembly and benefit subsequent modeling macroscopic models. Other robots will be configured as *free* robots and placed in the arena randomly and uniformly, similar to solute particles (see Figure 5). Then, the self-assembly formation will start automatically. The detailed control strategy of the robots in different states is introduced as follows.

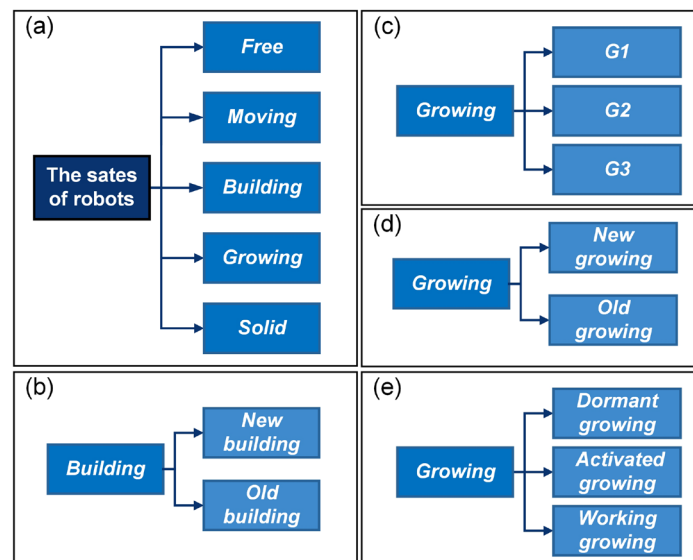


Figure 2. The definitions of robot's states. (a) The main states of robots (b) The sub-states of *building* robots depend on birth time. (c) The sub-states of *growing* robots depend on the number of neighboring filled unit cells. (d) The sub-states of *growing* robots depend on birth time. (e) The sub-states of *growing* robots depend on working modes.

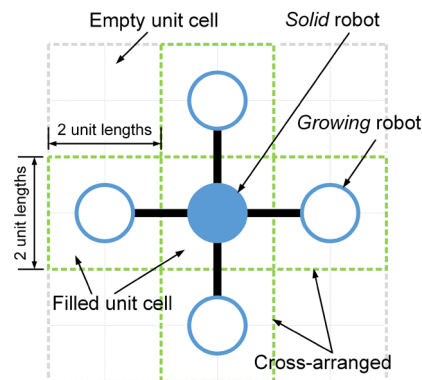


Figure 3. The cross-arranged unit cells for self-assembly formation inspired by Wigner–Seitz unit cell.

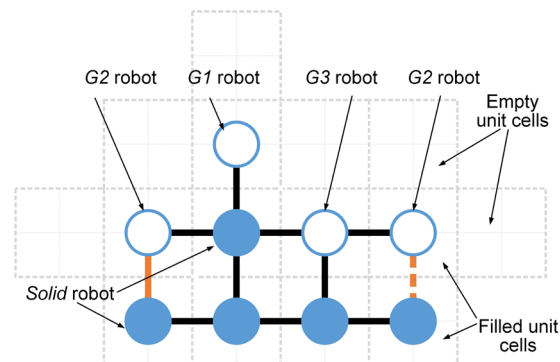


Figure 4. The sub-states of the growing state. Blue solid circle is the *Solid* robot. Blue hollow circle is the *growing* robot.

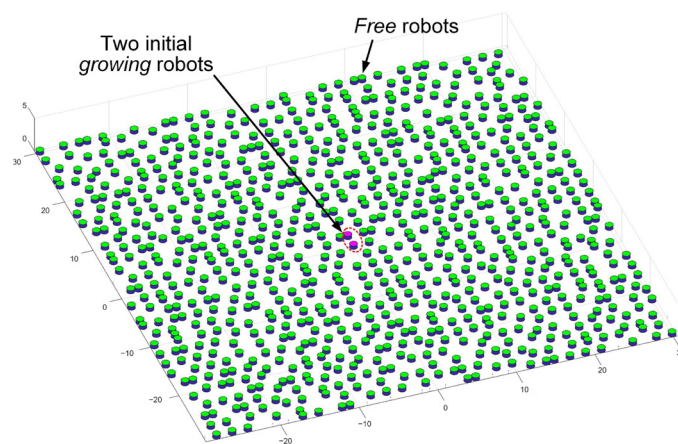


Figure 5. The initial conditions of the self-assembly formation and the specified initial *growing* robots inspired by nucleation. Two initial *growing* robots should be positioned at the coordinates (x_0, y_0) and $(x_0, y_0 + 2)$. x_0 and y_0 can be assigned any value. Here, (x_0, y_0) is set as $(0, 0)$.

When a *growing* robot is generated within a unit cell, its four neighboring unit cells that share an edge with its unit cell will be defined simultaneously (see Figure 3). Unit cells not yet filled by *growing* or *solid* robots are defined as empty; otherwise, they are defined as filled. Table 3 introduces the classifications and definitions of unit cells. Therefore, like the atoms on a growing crystal surface, the *growing* robots can recruit others to occupy their neighboring empty unit cells to expand the formation. Specifically, the *growing* robot selects one of the neighboring empty unit cells as an available empty unit cell. The selected one should be closest to the coordinate origin and not occupied by *building* robots. The coordinates of the available empty unit cell's center are then broadcast to recruit *free* robots.

As shown in Figure 6a, when the *growing* robot first broadcasts its selected coordinates at time step t , this broadcast only impacts the *free* robots. After receiving recruitments, the *free* robot turns into the *moving* state and selects the nearest available empty unit cell as its target position, moving towards it. During the move, the *moving* robot will not actively change its target position. To avoid self-assembly downtime, the moving time parameter T_M is defined. Thus, *moving* robots may turn into *free* robots due to exceeding the moving time parameter T_M . Furthermore, as the *growing* robots' broadcasts are public, multiple *moving* robots may hold the same target position. Hence, the competitive mechanism is applied based on the distance to the target position and ID number. As shown in Figure 6b, *moving* robots numbered 1, 3, and 6 share the same target position. The *moving* robot closest to the target position with the lowest ID number value can win the competition. Thus, in Figure 6b, the *moving* robots numbered 3 and 6 fail in the competition and will degenerate into *free* robots. The *moving* robot numbered 1 wins the competition and moves towards its target position. The *moving* robot that arrives at its target position will translate into a *building* robot (see Figure 6c). The *building* robots located in unit cell will wait for a few time steps (i.e., building time parameter T_B) to simulate the time-consuming of assembly and building work. Whereafter, they will become the *growing* robots, resulting in the expansion of the formation. If the currently selected available empty unit cell is occupied by a *building* robot, the *growing* robot will continue to choose a new available empty unit cell and broadcast for it. When all neighboring unit cells are filled unit cells, this robot completes its task and transitions from the *growing* state into the *solid* state. As shown in Figure 7, the finite-state machine (FSM) shows the whole control flow of the individual robot in the proposed self-assembly formation.

Table 3. Classifications and definitions of the unit cells.

Definition of Unit Cells	Descriptions
Filled unit cell	The unit cells are filled by <i>growing</i> or <i>solid</i> robots.
Empty unit cell	The unit cells are not filled by <i>growing</i> or <i>solid</i> robots.
Available empty unit cell	The empty unit cells selected by <i>growing</i> robots and are allowed as target positions for <i>free</i> and <i>moving</i> robots.
Building unit cell	The empty unit cells are allowed to be occupied by <i>building</i> robots.
Priority building unit cell	The building unit cells can be occupied by <i>building</i> robots preferentially.
Non-priority building unit cell	The building unit cells are occupied by <i>building</i> robots non-preferentially.

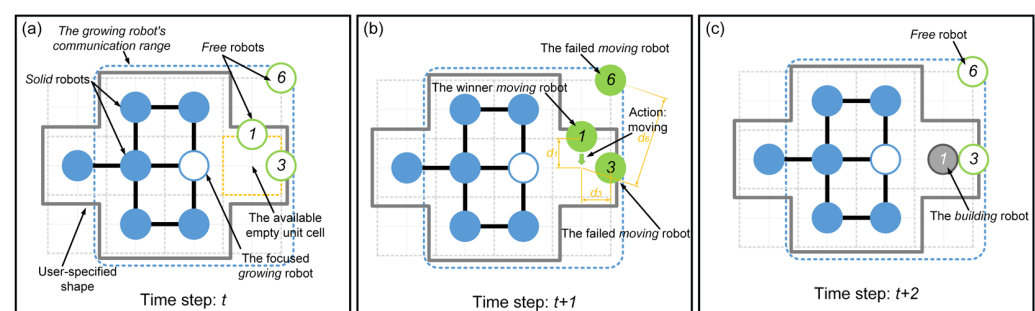


Figure 6. The competition mechanism of *moving* robots for the same target position. (a) Time step: t . (b) Time step: $t+1$. (c) Time step: $t+2$.

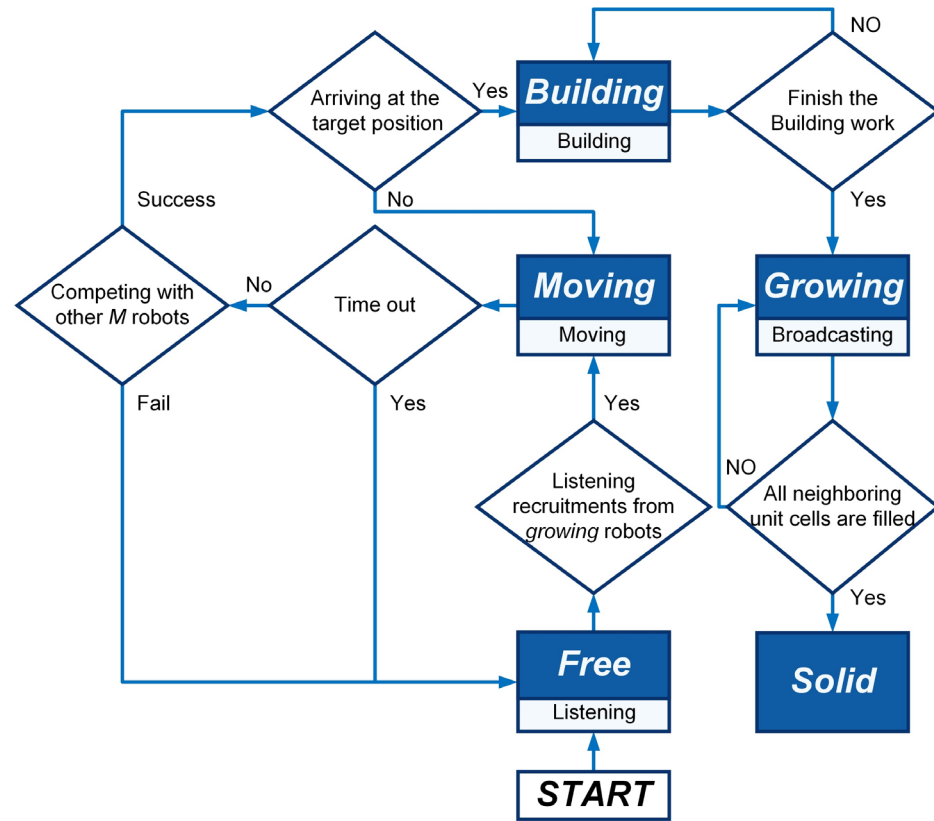


Figure 7. The FSM of a robot in the self-assembly formation.

2.3. The Implementation of the Self-Assembly Formation

As shown in Figures 8 and 9, the Waxberry robots are designed to achieve the proposed self-assembly formation, operating on a grid ground composed of unit square grids. The name “Waxberry” is derived from a unique fruit in China. As shown in Figure 8b, each Waxberry robot is equipped with two independent stepping motors. Under the control of the motor driver, each robot wheel can independently rotate forward or reverse at a specified speed. Thus, the Waxberry robot can move omnidirectionally based on the differential steering control. Here, the unit length and unit area are defined as the side length and area of a square unit grid, respectively. On the grid ground, the Waxberry robot can navigate between the locating points positioned on the nodes of each square unit grid (see Figure 9). Therefore, after obtaining its initial coordinates, the Waxberry robot can accurately determine its real-time position by tracking its trajectory using the principles of inertial navigation [59]. For simplicity, each movement from one locating point to any of its eight adjacent points is referred to as one time step. The local communication range of the Waxberry robot is limited to the area that the robot can reach in three time steps (see Figure 9). The information exchanged among Waxberry robots includes the robot’s ID number, state, present coordinates, and working information.

In this study, 15 Waxberry robots are manufactured. Each Waxberry robot is equipped with an identical self-assembly strategy, as described in Section 2.2. Two different shape formations are considered to test the proposed self-assembly strategy, i.e., the four-pointed star and the hexagonal crystal shape formations (see Figure 10). The system parameters of the self-assembly formation, i.e., building time parameter T_B and moving time parameter T_M , are set as one step and four steps by default. As shown in Figure 10, under the control of the proposed self-assembly formation strategy, 15 Waxberry robots autonomously assemble into 2 different shape formations without relying on external controls and the global leader. To verify the proposed self-assembly formation in a large-scale swarm, an embodied simulator capable of accommodating thousands of agents is developed. This simulator is programmed in MATLAB 2020, a high-level programming language designed

for engineers and scientists. Using MATLAB's powerful development capabilities, the simulator allows a large number of agents to simultaneously emulate the functionalities of the Waxberry robot. Each agent in the simulator can move, communicate, and make autonomous decisions. Consequently, the proposed self-assembly can be further tested at large-scale formations. Here, the formation scale is expanded to include about 300 robots, 1500 robots, and 3500 robots, respectively. Based on the proposed self-assembly strategy, swarm robotics can successfully form the two considered shape formations. Figure 11 shows the self-assembly process when the formation is composed of about 300 robots.

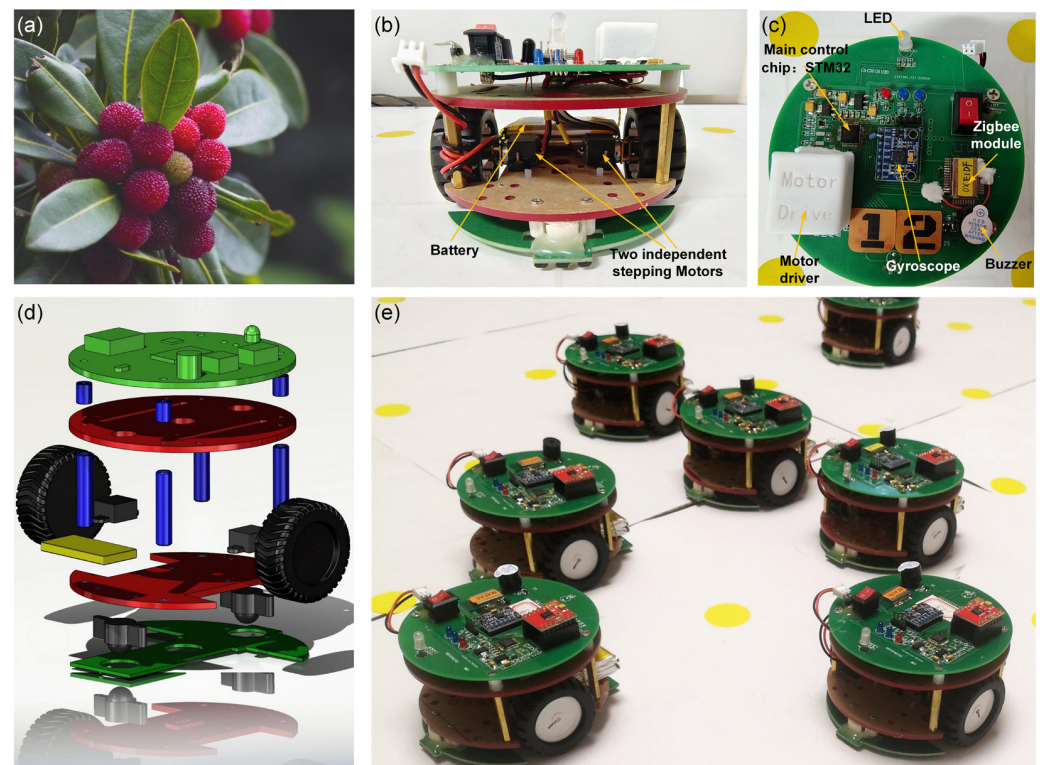


Figure 8. Waxberry and Waxberry robots. (a) Waxberry. (b) The battery and motors of the Waxberry robot. (c) The electronic system of the Waxberry robot. (d) The 3D design of the Waxberry robot. (e) The swarm robotics is composed of multiple Waxberry robots.

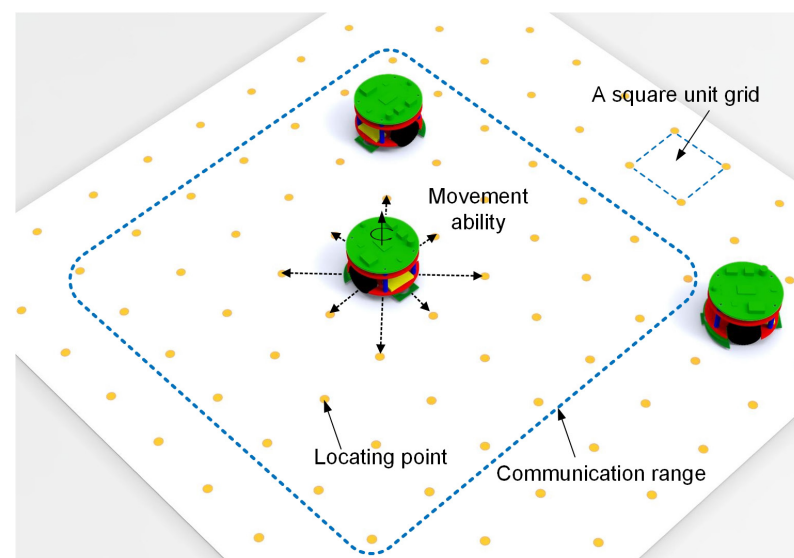


Figure 9. The foundational ability of the Waxberry robot.

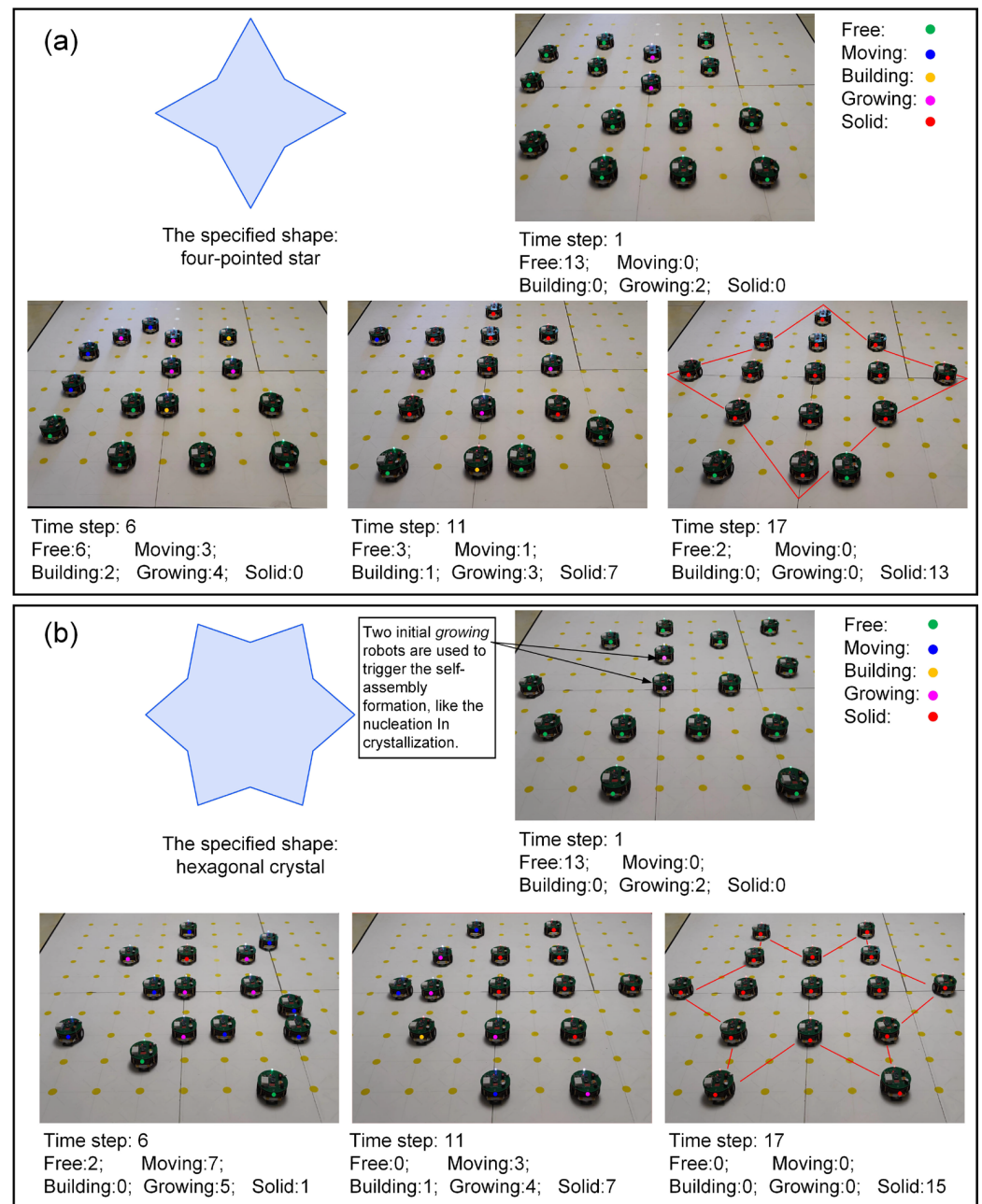


Figure 10. The self-assembly formation achieved by 15 Waxberry robots. (a) Self-assembly of a four-pointed star shape formation. (b) Self-assembly of a hexagonal crystal shape formation.

The above real-robot experiments and simulations demonstrate the proposed self-assembly strategy's high flexibility, scalability, and efficiency. The robots in different states can self-assemble into various shape formations through local interactions and cooperation, demonstrating flexibility in different shape conditions. Meanwhile, compared to the works in Refs. [29–31], the proposed distributed self-assembly formation does not rely on a global leader and pre-design global information. This allows swarm robotics more flexible deployment capabilities, showing massive potential in practical applications. Additionally, tests conducted at the formation scale, ranging from about 15 to 3500 robots, verify that the proposed self-assembly strategy exhibits excellent scalability. Such scalability comes from a fully distributed self-assembly strategy based on local interactions. That is, the local environments of individual robots are similar, no matter how large the scale of the swarm robotics. High forming efficiency is also one of the advantages of the proposed self-assembly formation. Compared with the one or two parallel motion chains in Refs. [25–27],

the proposed self-assembly formation shows remarkable parallelism capability, as all *growing* robots contribute to the expansion of the formation. Indeed, as the formation grows larger, the forming efficiency becomes higher due as there are more *growing* robots involved in the construction. Thus, the proposed self-assembly formation shows a rare and precious super-linear feature. This feature is analyzed in detail below.

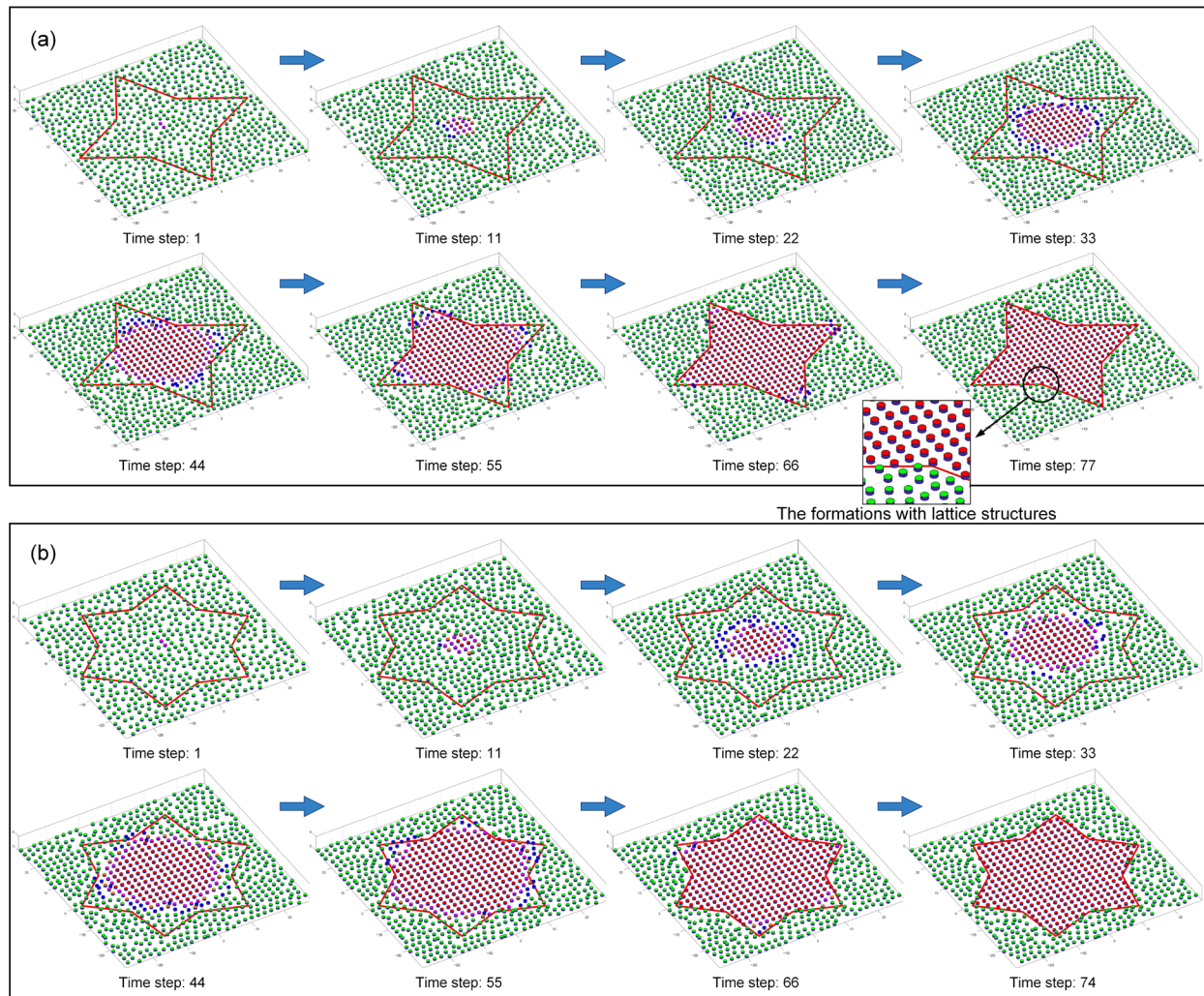


Figure 11. Self-assembly a formation comprising about 300 robots. (a) Self-assembly a four-pointed star shape formation. (b) Self-assembly a hexagonal crystal shape formation. Green dots: *free* robots. Blue dots: *moving* robots. Orange dots: *building* robots. Pink dots: *growing* robots. Red dots: *solid* robots.

The super-linear feature is strong evidence for the high forming efficiency. That is, the forming efficiency of the self-assembly increases with the scale of the formation [13]. It is a rare and precious feature. The efficiency of the existing self-assembly formation designed by Rubenstein et al. [25] and Yang et al. [26] remains constant and does not increase as the formation scale becomes larger (see Figure 12). To analyze the super-linear feature, the average number of robots joining the formation within one time step is defined as the forming efficiency index e_c . It can be expressed as follows.

$$e_c = \frac{m_0}{T} \quad (1)$$

here, T represents the total time dedicated to the self-assembly formation, while m_0 denotes the number of robots positioned within the formation (i.e., the formation scale). As shown

in Figure 12, whether forming the four-pointed star shape or the hexagonal crystal shape formation, the simulation results show that the forming efficiency increases with the scale of the formation. In the small-scale formation comprising about 15 robots, the efficiencies e_c of forming the four-pointed star shape formation and the hexagonal crystal shape formation are 0.764 and 0.598, respectively. However, when the scale formation is expanded to about 3500 robots, the efficiencies e_c are 13.418 and 14.455, respectively. The efficiencies of forming the four-pointed star shape and the hexagonal crystal shape formations are increased by 1756% and 2417%, as the formation scale increases from about 15 to 3500 robots. The efficiencies e_c of self-assembly formation designed by Rubenstein et al. [25] and Yang et al. [26] can be viewed as one and two, respectively, using robots of the same performance without considering the preparation time. Although our proposed self-assembly formation cannot show an efficiency advantage over the work in Refs. [25,26], the efficiency of the proposed self-assembly formation is greater than 2 when more than 100 robots form the formation (see Figure 12). In the formation comprising about 3500 robots, the efficiency of the proposed self-assembly formation is 13.9 and 6.97 times more efficient than the studies of Rubenstein et al. [25] and Yang et al. [26], respectively.

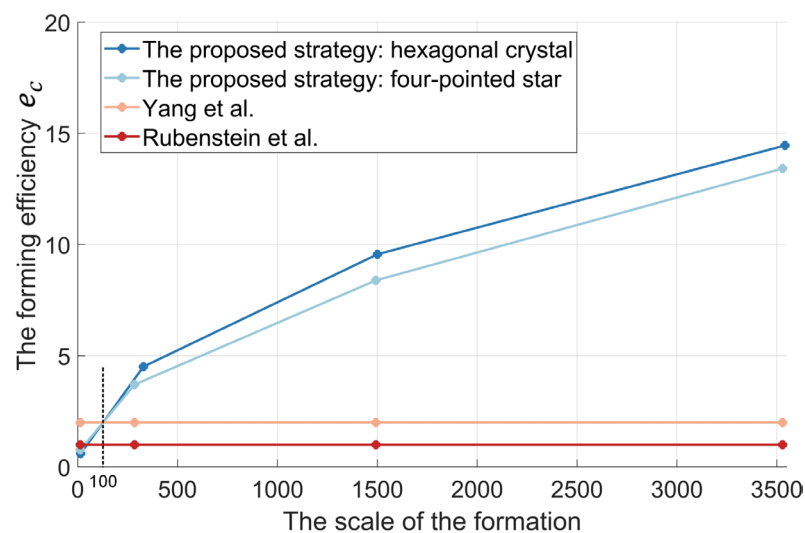


Figure 12. The super-linear of the proposed self-assembly formation based on simulation results [25,26].

3. Mathematical Description of the Macroscopic Model

The macroscopic model is an important mathematical tool for predicting and studying swarm behaviors. This section develops a non-spatial macroscopic model for the proposed distributed self-assembly formation. The master difference equations (i.e., the model's framework) are described in Section 3.1. The analysis for the structural features of the shape formation is introduced in Section 3.2. Subsequently, all state transition functions in this macroscopic model are estimated in Section 3.3 according to the structural features. Lastly, the effects of the shape boundary's constraints are shown in Section 3.4.

3.1. The Master Difference Equations

In this section, the non-spatial macroscopic model is developed to study the characteristics of the proposed self-assembly formation's swarm behaviors. Referring to Ref. [46], a stock and flow diagram (see Figure 13) is created to depict the swarm behaviors of the proposed self-assembly formation, utilizing the FSM depicted in Figure 7. The stocks, represented by boxes in Figure 13, denote the average number of robots in various predefined states at the macroscopic level. Arrows in Figure 13 depict flows, expressing the number of robots transitioning between states. Such changes can be described by state transition functions. Based on this stock and flow diagram, a set of difference equations in the discrete-time domain can be derived to model the fluctuations in the average number

of robots between different states. The framework of the macroscopic model is outlined as follows.

$$N_F(t+1) = N_F(t) - \Gamma_F(t) + \Gamma_M^F(t) + \Lambda_M^F(t, T_M) \quad (2)$$

$$N_M(t+1) = N_M(t) - \Gamma_M^F(t) - \Gamma_M^B(t) - \Lambda_M^F(t, T_M) + \Gamma_F(t) \quad (3)$$

$$N_B(t+1) = N_B(t) - \Lambda_B^G(t, T_B) + \Gamma_M^B(t) \quad (4)$$

$$N_{G1}(t+1) = N_{G1}(t) + \Lambda_B^{G1}(t, T_B) - \Delta_{G1}^{G2}(t) - \Delta_{G1}^{G3}(t) - \Omega_{G1}(t) \quad (5)$$

$$N_{G2}(t+1) = N_{G2}(t) + \Lambda_B^{G2}(t, T_B) + \Delta_{G1}^{G2}(t) - \Delta_{G2}^{G3}(t) - \Omega_{G2}(t) \quad (6)$$

$$N_{G3}(t+1) = N_{G3}(t) + \Delta_{G1}^{G3}(t) + \Delta_{G2}^{G3}(t) - \Delta_{G3}^S(t) \quad (7)$$

$$N_S(t+1) = N_S(t) + \Delta_{G3}^S(t) + \Omega_{G1}(t) + \Omega_{G2}(t) \quad (8)$$

As shown in Table 4, $N_F(t)$, $N_M(t)$, $N_B(t)$, and $N_S(t)$ are the average number of robots in state *free*, *moving*, *building*, and *solid* respectively at time step t . $N_{G1}(t)$, $N_{G2}(t)$, and $N_{G3}(t)$ represent the average number of robots in state G1, G2, and G3 at time step t . The definitions of the G1, G2, and G3 states can be found in Section 2.2 and Figure 4. Furthermore, the total number of *growing* robots is expressed as $N_G(t)$.

Table 4. Description of notations in the master difference equations.

Notations	Descriptions
$N_F(t)$	The number of <i>free</i> robots at time step t .
$N_M(t)$	The number of <i>moving</i> robots at time step t .
$N_B(t)$	The number of <i>building</i> robots at time step t .
$N_G(t)$	The number of <i>growing</i> robots at time step t .
$N_{G1}(t)$	The number of G1 robots at time step t .
$N_{G2}(t)$	The number of G2 robots at time step t .
$N_{G3}(t)$	The number of G3 robots at time step t .
$N_S(t)$	The number of <i>solid</i> robots at time step t .
$\Gamma_F(t)$	The number of <i>free</i> robots becoming <i>moving</i> robots at time step t due to receiving broadcasts.
$\Gamma_M^F(t)$	The number of <i>moving</i> robots becoming <i>free</i> robots at time step t due to failure in competition.
$\Lambda_M^F(t, T_M)$	The number of <i>moving</i> robots becoming <i>free</i> robots at time step t due to running out of the moving time parameter T_M .
$\Gamma_M^B(t)$	The number of <i>moving</i> robots becoming <i>building</i> robots as they succeed in competition and arrive at their target positions at time step t .
$\Lambda_B^G(t, T_B)$	The number of <i>building</i> robots becoming <i>growing</i> robots at time step t after spending T_B steps assembling the formation.
$\Lambda_B^{G1}(t, T_B)$	The number of <i>building</i> robots becoming G1 robots at time step t after spending T_B steps assembling the formation.
$\Lambda_B^{G2}(t, T_B)$	The number of <i>building</i> robots becoming G2 robots at time step t after spending T_B steps assembling the formation.
$\Delta_{G1}^{G2}(t)$	The number of G1 robots becoming G2 robots at time step t .
$\Delta_{G1}^{G3}(t)$	The number of G1 robots becoming G3 robots at time step t .
$\Delta_{G2}^{G3}(t)$	The number of G2 robots becoming G3 robots at time step t .
$\Delta_{G3}^S(t)$	The number of G3 robots becoming <i>solid</i> robots at time step t .
$\Omega_{G1}(t)$	The number of G1 robots becoming <i>solid</i> robots at time step t due to constraints of shape boundary.
$\Omega_{G2}(t)$	The number of G2 robots becoming <i>solid</i> robots at time step t due to constraints of shape boundary.

The state transition functions describe the changes in the number of robots transitioning between different states at time step t . The details are introduced as follows. $\Gamma_F(t)$ represents the number of *free* robots that turn into the *moving* state upon receiving broadcasts from *growing* robots. $\Gamma_M^F(t)$ represents the number of robots in the *moving* state that degenerate into the *free* state due to failure in competition with other *moving* robots for

the same target position. $\Gamma_M^B(t)$ represents the number of *moving* robots that succeed in competition and arrive at their target positions. $\Lambda_M^F(t, T_M)$ denotes the number of *moving* robots that run out of the moving time parameter T_M and become the *free* state. $\Lambda_B^{G1}(t, T_B)$ and $\Lambda_B^{G2}(t, T_B)$ represent the number of *building* robots that transition into G1 and G2 states, respectively, after spending T_B steps assembling the formation. Their total number is expressed as $\Lambda_B^G(t, T_B)$. When the *building* robots occupying the empty unit cells become the *growing* robots, the existing *growing* robots will update their sub-states or translate into the *solid* state. These transitions are denoted as $\Delta_{G1}^{G2}(t)$, $\Delta_{G1}^{G3}(t)$, $\Delta_{G2}^{G3}(t)$, and $\Delta_{G3}^S(t)$, respectively. Additionally, $\Omega_{G1}(t)$ and $\Omega_{G2}(t)$ signify the count of G1 and G2 robots directly transitioning into *solid* robots due to the constraints of the shape boundary, respectively. At this point, the framework of the proposed macroscopic model is described completely.

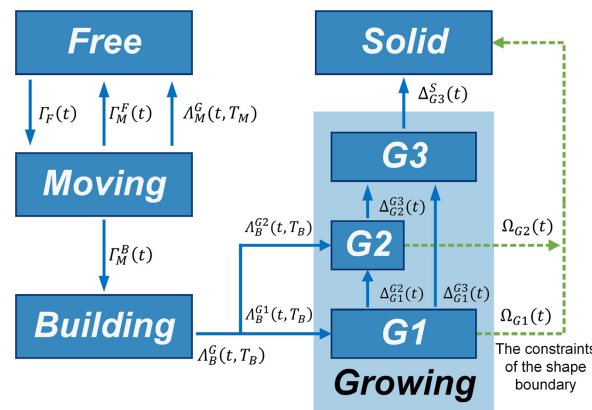


Figure 13. The stock and flow diagram of the proposed macroscopic model.

3.2. The Analysis for the Structural Features

In the proposed self-assembly formation, all *growing* robots collectively constitute the growth boundary of the formation. The arrangement of *growing* robots based on the cross-arranged unit cells inevitably creates intricate structures within the growth boundary, causing strong nonlinearity for the state transitions. As shown in Table 5, the influence of spatial structures renders traditional geometrical estimation methods ineffective in deducing corresponding state transition functions. Similarly, the data statistics method falls short in elucidating the internal mechanisms of state transitions. Consequently, analyzing the structural features within the growth boundary of the formation is an essential precondition for constructing state transition functions capable of probing into the mechanisms of swarm behaviors. Although the macroscopic model considers the features of spatial structures, it remains non-spatial as the probability of these structural features affecting each robot is uniform. Here, the definition and quantities of six basic structural types are discussed to depict the structural features of the growth boundary. Subsequently, the distribution characteristics of *building* robots within the empty unit cells provided by different structural types are discussed. The arrangement characteristics of the structural types are also analyzed. Note that the distribution characteristics of *building* robots in empty unit cells and the arrangement characteristics of structural types are regarded as independent in probability. Additionally, some assumptions for the macroscopic model should be elucidated to facilitate the discussion of structural features. As introduced in Section 2.2, all robots share the same ability and an identical self-assembly strategy. In the task of the self-assembly formation, all robots distributed across the arena randomly and uniformly will experience *free*, *moving*, *building*, *growing*, and *solid* states and perform corresponding tasks to complete the self-assembly. Therefore, based on the principle of equalitarianism and the actual situations of self-assembly formation, the macroscopic model should satisfy the following assumptions:

Assumption 1. The building efficiency of any unit cell remains constant both in terms of time and space.

Assumption 2. The formation expands at a consistent rate in all directions.

Assumptions 1 and 2 are consistent with the fact that the model is non-spatial. In the structural features analysis, these assumptions can reduce the number of situations that need to be discussed (see Section 3.2) and simplify the complexity of the model. Note that the above assumptions are only for the self-assembly of swarm robotics and have no bearing on crystallization.

Table 5. The comparison among the existing methods and our proposed method.

	Applicable to Spatial Structure Scenarios	Explaining the Inner Mechanisms of the State Transitions
Data statistics method [50–53]	✓	×
Geometrical estimation method [14,16]	×	✓
Structural feature estimation method	✓	✓

3.2.1. The Definitions of Six Basic Structural Types

As shown in Figure 14, six basic structural types are defined based on $G2$ robots. Each structural type comprises only one $G2$ robot, positioned on the leftmost side of the structural type. Thus, the total count of structural types corresponds to the number of $G2$ robots. The structural types consisting of only $G2$ and $G3$ robots can be classified as Fa and Ra based on whether the angle between $G2$ and its neighboring *growing* robot to the right is 180° (flat angle) or 90° (right angle) (see Figure 14a,c). When the structural type Fa contains a $G3$ robot, it is subdivided into Fa_2 ; otherwise, it is Fa_1 . Similarly, the structural type Ra is also divided into Ra_1 and Ra_2 . The situations involving structural types that include more than one $G3$ robot are not considered in this paper since the $G3$ robot builds its neighboring empty unit cells with the same efficiency as other *growing* robots (see Assumption 1). To simplify the analysis, $G1$ robots are only considered within structural types Fa , i.e., structural types Fa_1 and Fa_2 (see Figure 14e,f). That is, we ignored some of the rare structure types without reducing the accuracy of the model predictions. Using these defined six basic structural types, the growth boundary's structural feature can be described precisely. The notations related to all state transition functions are explained in Table 6.

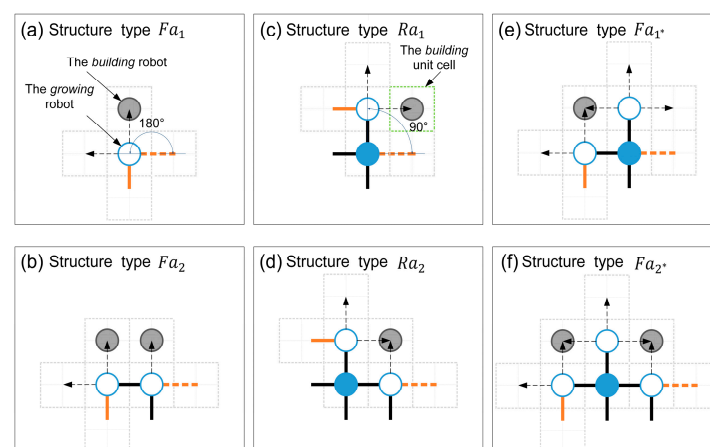


Figure 14. Six basic structural types. The orange heavy solid line and orange dotted line represent the connection interfaces to the previous and next structural types, respectively. The black dotted arrows points to the neighboring empty unit cells of the *growing* robots. The green dotted boxes represent the building unit cells, which are the empty unit cells allowed to be occupied by the *building* robots. Hollow blue circles represent *growing* robots, solid blue circles represent *solid* robots, and solid gray circles represent *building* robots.

Table 6. Description of notations in all state transition functions.

Notations	Descriptions
$N_{Fa}(t)$	The number of structural types Fa .
$N_{Fa_1}(t)$	The number of structural types Fa_1 .
$N_{Fa_1^*}(t)$	The number of structural types Fa_1^* .
$N_{Fa_2}(t)$	The number of structural types Fa_2 .
$N_{Fa_2^*}(t)$	The number of structural types Fa_2^* .
$N_{Ra}(t)$	The number of structural types Ra .
$N_{Ra_1}(t)$	The number of structural types Ra_1 .
$N_{Ra_2}(t)$	The number of structural types Ra_2 .
$N_{bu}(t)$	The total number of building unit cells.
$N_{bu_p}(t)$	The total number of the priority building unit cells.
$N_{B_{new}}(t)$	The number of <i>new building</i> robots.
$N_{B_{old}}(t)$	The number of <i>old building</i> robots.
σ_{old}	The probability of <i>old building</i> robots in the priority building unit cell.
ϵ_{old}	The probability of <i>old building</i> robots in the non-priority building unit cell.
σ_{new}	The probability of <i>new building</i> robots in the priority building unit cell.
ϵ_{new}	The probability of <i>new building</i> robots in the non-priority building unit cell.
τ	The probability of structural type Ra_1 being on the left side of structural types Ra_2 and Fa .
τ^*	The probability of structural type Ra_1 being on the left side of structural type Ra_1 .
μ_{Ra_1}	The probability of the <i>building</i> robots in the building unit cell provided by the G2 robots of the structural type Ra_1 .
$\mu_{Ra_1}^{old}$	The probability of the <i>old building</i> robots in the building unit cell provided by the G2 robots of the structural type Ra_1 .
$\mu_{Ra_1}^{new}$	The probability of the <i>new building</i> robots in the building unit cell provided by the G2 robots of the structural type Ra_1 .
μ_{Ra_2}	The probability of the <i>building</i> robots in the building unit cell provided by the G2 robots of the structural type Ra_2 .
μ_{Fa}	The probability of the <i>building</i> robots in the building unit cell provided by the G2 robots of the structural type Fa .
μ_{Fa}^{old}	The probability of the <i>old building</i> robots in the building unit cell provided by the G2 robots of the structural type Fa .
μ_{Fa}^{new}	The probability of the <i>new building</i> robots in the building unit cell provided by the G2 robots of the structural type Fa .
ρ_{G1}^{new}	The probability of the G1 robots in a <i>new growing</i> state.
ρ_{G2}^{new}	The probability of the G2 robots in a <i>new growing</i> state.
σ_{Λ_B}	The probability of the robot transitioning from <i>building</i> to <i>growing</i> state within the priority building unit cells.
ϵ_{Λ_B}	The probability of the robot transitioning from <i>building</i> to <i>growing</i> state within the non-priority building unit cells.

3.2.2. The Calculation of the Quantities of Six Basic Structural Types

According to **Assumptions 1** and **2**, the shape of the formation's growth boundary must be a convex polygon before the shape boundary constraint is triggered. As shown in Figure 15, the structural types Fa can bend the extension of the growth boundary by 90° in addition to increasing the length of the growth boundary. Therefore, based on the basic geometric principles, a growth boundary allows for only four structural types of Fa . The growth boundary can be understood as composed of four equal-length growth edges originating from the structural type Fa . The simplest growth edge contains only a structural-type Fa (see Figure 15a). Thus, the number of structural-type Fa is written as $N_{Fa}(t)$, i.e.,

$$N_{Fa}(t) = \min(4, N_{G2}(t)) \quad (9)$$

where $N_{G2}(t)$ expresses the number of G2 robots. The number of structural types of Fa_1 , Fa_1^* , Fa_2 , and Fa_2^* at time step t can be expressed as $N_{Fa_1}(t)$, $N_{Fa_1^*}(t)$, $N_{Fa_2}(t)$, and $N_{Fa_2^*}(t)$, i.e.,

$$N_{Fa_1}(t) = N_{Fa}(t)(1 - \rho_{G3})(1 - \rho_{G1}) \quad (10)$$

$$N_{Fa_1^*}(t) = N_{Fa}(t)(1 - \rho_{G3})\rho_{G1} \quad (11)$$

$$N_{Fa_2}(t) = N_{Fa}(t)\rho_{G3}(1 - \rho_{G1}) \quad (12)$$

$$N_{Fa_2^*}(t) = N_{Fa}(t)\rho_{G3}\rho_{G1} \quad (13)$$

where ρ_{G1} is the probability of *G1* robots in structural type *Fa*, which is defined as

$$\rho_{G1} = \frac{N_{G1}(t)}{N_{Fa}(t)} \quad (14)$$

The ρ_{G3} denotes the probability that structural types *Fa* and *Ra* contain a *G3* robot. Due to equalitarianism, ρ_{G3} can be expressed as

$$\rho_{G3} = \frac{N_{G3}(t)}{N_{G2}(t)} \quad (15)$$

Here, $N_{G2}(t)$ represents the total number of all structural types. The number of the structural type *Ra* can be written as $N_{Ra}(t)$, i.e.,

$$N_{Ra}(t) = N_{G2}(t) - N_{Fa}(t) \quad (16)$$

Similarly, the number of structural types *Ra*₁ and *Ra*₂ is represented as follows.

$$N_{Ra_1}(t) = N_{Ra}(t)(1 - \rho_{G3}) \quad (17)$$

$$N_{Ra_2}(t) = N_{Ra}(t)\rho_{G3} \quad (18)$$

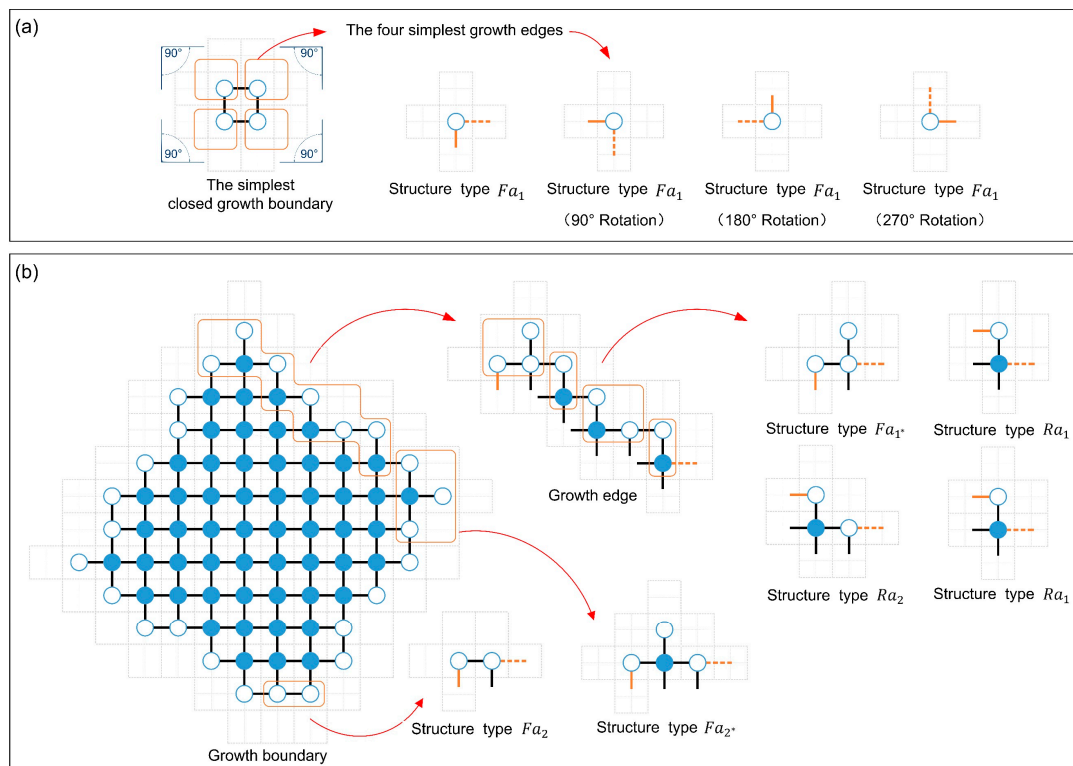


Figure 15. The relationship between growth boundary, growth edge, and structural types. (a) The disassembly of the simplest growth boundary. (b) The disassembly of a normal growth boundary.

3.2.3. The Distribution Characteristics of Building Robots

The distribution of *building* robots within the empty unit cells provided by the structural types should be discussed since it plays a crucial role in the state transitions of *building*

and *growing* robots. The empty unit cells allowed to be occupied by the *building* robots are considered the building unit cells (see Table 3 and Figure 14). Therefore, the distribution rules and classification of the building unit cells are determined first. Then, the sub-states of *building* robots should be discussed further. Finally, the probability of different *building* robots in different building unit cells can be determined.

As a prerequisite for estimating the number of building unit cells, based on a large number of simulations and experimental results, the following two phenomena related to empty unit cells deserve attention. Firstly, the building time parameter of *building* robots considered in this work is very small (i.e., $T_B \leq 4$), usually less than the time required for the robot to transition from *free* to *building* state. Therefore, in the case of a *growing* robot with multiple empty unit cells (such as *G1* and *G2* robots), when it autonomously recruits a *building* robot for the second empty unit cell, the *building* robot within its first empty unit cell has already transitioned to the *growing* state due to completing its assembly works. As this transition also triggers the original *growing* robot to undergo a state transfer, it is impossible for a *growing* robot to actively recruit multiple *building* robots simultaneously without triggering its state transition. Secondly, the empty unit cell shared by two neighboring *growing* robots is typically closer to the coordinate origin than its neighbor. According to the self-assembly strategy, *growing* robots prioritize selecting empty unit cells of this kind and broadcast their coordinates to recruit *free* and *moving* robots. Consequently, empty unit cells shared by two adjacent *growing* robots are more likely to be occupied by *building* robots, thereby becoming the building unit cells. Under the premise of satisfying the above phenomena, the simplified distribution rules for building unit cells are defined as follows to minimize model complexity.

- (I) Each *growing* robot is deemed to actively provide only one empty unit cell as the building unit cell.
- (II) The *G1* robot must share the building unit cell with the *G2* robot in the current structural type.
- (III) The building unit cells actively provided by the *G2* robot should be unaffected by the structural types on the left.
- (IV) *G2* robots preferentially select empty unit cells shared with adjacent *growing* robots of the same structural types as the building unit cell.

Here, the total number of building unit cells can be written as follows.

$$N_{bu}(t) = N_{Fa_1}(t) + 2N_{Fa_2}(t) + N_{Fa_{1*}}(t) + 2N_{Fa_{2*}}(t) + N_{Ra_1}(t) + N_{Ra_2}(t) \quad (19)$$

The green dotted boxes in Figure 14 represent the possible building unit cells. Note that a *building* robot cannot become a *G3* robot directly under current situations.

Indeed, a particular type of building unit cell, i.e., a priority building unit cell, is given preference for occupation by the *building* robots. According to the self-assembly strategy, once a *building* robot occupies a neighboring empty unit cell, the *growing* robot will immediately broadcast the next suitable neighboring unit cell's coordinates. It is not hard to see that the corresponding building unit cell's coordinates have been broadcast before a *growing* robot is upgraded to the *G3* sub-state. Consequently, the building unit cells provided by *G3* robots are designated priority building unit cells. In structural types Fa_{1*} and Fa_{2*} , the simultaneous birth of two neighboring *G1* and *G2* robots from the *building* state is typically not feasible. Thus, when a *G1* robot sits next to a *G2* robot, their sharing building unit cell has been broadcast and should be treated as the priority. Besides the proposed self-assembly strategy, the generation of priority building unit cells is also related to the autonomous actions of robots. Estimating the priority building unit cell's number based only on the self-assembly strategy and structural features is prone to bias. Considering the randomness of autonomous actions of robots, it is assumed that there are ζ priority building unit cells existing in structural types Ra_1 to correct the estimate of the priority building unit cell's number. Note that the autonomous actions of robots are affected by the external environment, i.e., the proportion of robots in different states within the

swarm. As defined in Section 2.2, the building time parameter T_B can change the survival duration of the *building* and *growing* robots to change the proportion of robots in different states within the swarm. Thus, based on the statistical result of simulations, Table 7 shows the values of ζ under different building time parameters T_B . The total number of the priority building unit cells is written as follows.

$$N_{bu_p}(t) = N_{Fa_1^*}(t) + 2N_{Fa_2^*}(t) + N_{Fa_2}(t) + N_{Ra_2}(t) + \zeta \quad (20)$$

Table 7. The value of coefficients ζ and φ .

Building Time Parameter T_B (Unit: Steps)	1	2	3	4	≥ 5
ζ	1.3	1.1	0.7	0.5	0.2
φ	0.62	0.74	0.76	0.78	0.8

ζ can be found in Equation (20). φ can be found in Equation (54).

Other building unit cells are considered non-priority building unit cells, except for priority building unit cells.

Additionally, it is imperative to delve deeper into the sub-states of the *building* robot. These sub-states can be used to distinguish the neighboring *growing* robot's working modes. This differentiation is crucial for constructing the state transition functions $\Gamma_M^F(t)$. Here, the *building* robots transitioning from the *moving* robot at time step $t - 1$ are referred to as the *new building* robots, while others are classified as *old building* robots (see Figure 2b). According to **Assumption 1** and the discussion of the priority building unit cells, *old building* robots are given preference for occupying the priority building unit cells compared to *new building* robots. Thus, the probability of *old building* robots being distributed in the priority building unit cell is

$$\sigma_{old} = \min(1, \frac{N_{B_{old}}(t)}{N_{bu_p}(t) + e_0}) \quad (21)$$

where e_0 is a value of 10^{-5} to avoid miscalculation. The $N_{B_{old}}(t)$ expresses the number of *old building* robots at time step t , which can be written as follows.

$$N_{B_{old}}(t) = N_B(t) - N_{B_{new}}(t) \quad (22)$$

where $N_{B_{new}}(t)$ is the number of *new building* robots at time step t , which is equal to $\Gamma_M^B(t - 1)$ (see Equation (3)). The probability that the *old building* robots are distributed in the non-priority building unit cell is

$$\epsilon_{old} = \frac{N_{B_{old}}(t) - \sigma_{old}N_{bu_p}(t)}{N_{bu}(t) - N_{bu_p}(t) + e_0} \quad (23)$$

The probabilities of the *new building* robot distributed in the priority building unit cell and the non-priority building unit cell are

$$\sigma_{new} = \min(1 - \sigma_{old}, \frac{N_{B_{new}}(t)}{N_{bu_p}(t) + e_0}) \quad (24)$$

$$\epsilon_{new} = \frac{N_{B_{new}}(t) - \sigma_{new}N_{bu_p}(t)}{N_{bu}(t) - N_{bu_p}(t) + e_0} \quad (25)$$

3.2.4. The Arrangement Characteristics of Six Basic Structural Types

The arrangement of these structural types is another key factor affecting the state transitions. Note that the arrangement characteristics of the structural types and the

distribution characteristics of the *building* robots are probabilistically independent of each other. According to the analysis of *building* robot distributions within building unit cells, the structural type Ra_1 plays a unique role. As shown in Figure 16, when a *growing* robot is born in the building unit cell provided by the structural type Ra_1 , both the G2 robot in the current structural type Ra_1 and another G2 robot in the right structural type will update to G3 robots simultaneously. That is, the structural type Ra_1 can affect its adjacent right structural types. This is a special feature that other structural types don't have. Therefore, the key of the arrangement analysis is to estimate the probability that the left side of the focused structural type is structural type Ra_1 . Here, based on the principle of equalitarianism, all structural types Ra_1 are evenly allocated to four growth edges and are arranged after the starting structural type Fa . Then, the structural types Ra_2 have an equal probability of appearing in all vacancies generated by structural types Fa and Ra_1 (see Figure 17). Utilizing permutation and combination theory [60], we can calculate the probability of structural type Ra_1 being on the left side of structural types Ra_2 and Fa as follows.

$$\tau = \min \left(1, \frac{N_{Ra_1}(t)}{N_{Fa}(t) + N_{Ra_2}(t)} \right) \quad (26)$$

where $N_{Fa}(t) + N_{Ra_2}(t)$ represents total number of the vacancies generated by structural types Ra_2 and Fa . The probability that a structural type Ra_1 is on the left side of another structural type Ra_1 is

$$\tau^* = \frac{\max(0, N_{Ra_1}(t) - N_{Fa}(t) - N_{Ra_2}(t))}{N_{Fa}(t) + N_{Ra_2}(t)} \quad (27)$$

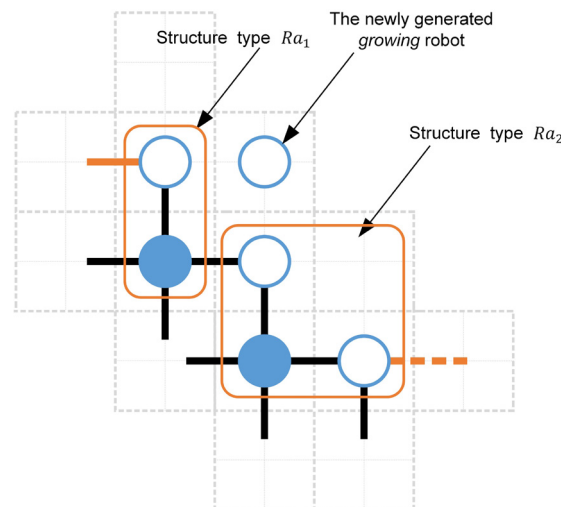


Figure 16. The effect of the structural type Ra_1 for other structural types.

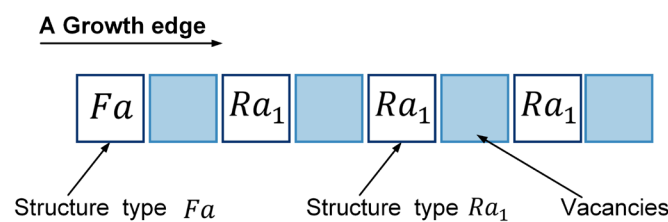


Figure 17. The arrangement approach of structural types within a growth edge.

3.3. The State Transition Functions Based on the SFE

3.3.1. The State Transition Function $\Gamma_F(t)$

The $\Gamma_F(t)$ represents the number of *free* robots transitioning to the *moving* state due to receiving recruitments of the *growing* robots. It is written as follows.

$$\Gamma_F(t) = S_0 d_0 - N_M(t) - N_B(t) \quad (28)$$

where S_0 represents the total communication area created by all *growing* robots. Note that the communication range of a *growing* robot can also cover the formed formation. This area will not be taken into consideration for S_0 (see Figure 18). Within this communication area, d_0 denotes the robot's density, which is defaulted to 0.2 (see Section 2.2). $S_0 d_0$ represents the total number of robots. Furthermore, all *moving* and *building* robots are situated within the total communication area, enabling interaction with *growing* robots. Therefore, $\Gamma_F(t)$ can be expressed by Equation (28).

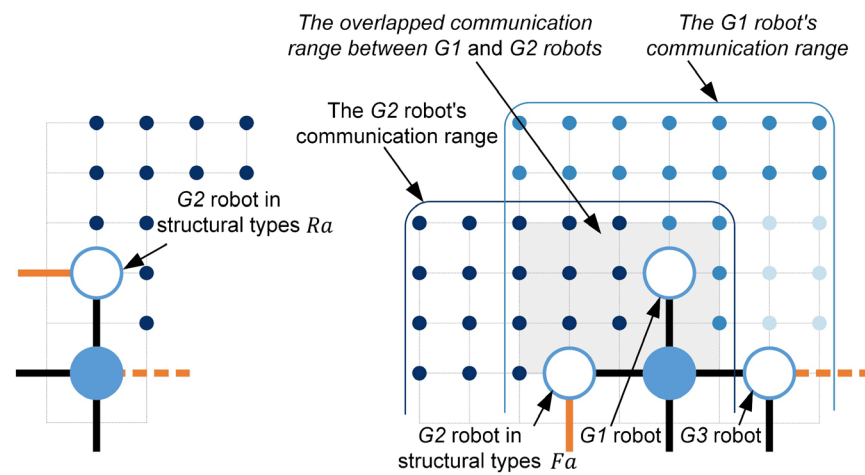


Figure 18. The definition of the independent communication ranges of *growing* robots. Blue solid dots: G1 robot's independent communication ranges. Dark blue solid dots: G2 robot's independent communication ranges. Light blue solid dots: G3 robot's independent communication ranges.

The total communication area S_0 created by all *growing* robots is the key to constructing the state transition function $\Gamma_F(t)$. As shown in Figure 9, the communication range of a robot can be regarded as 7×7 position points. Due to the defined cross-arrangement, there will be some overlap in the communication ranges of two neighboring *growing* robots (see Figure 19). To avoid repetitive computation, Equation (29) is proposed to calculate the total communication area based on the structural types. It can be written as follows.

$$S_0 = \alpha_1 N_{Fa_1}(t) + \alpha_2 N_{Fa_1^*}(t) + \alpha_3 N_{Fa_2}(t) + \alpha_4 N_{Fa_2^*}(t) + \alpha_5 N_{Ra_1}(t) + \alpha_6 N_{Ra_2}(t) - \alpha_0 D_{G2} \quad (29)$$

To calculate the coefficients α_0 to α_6 , the independent communication ranges of *growing* robots that do not overlap are defined artificially. Solid dots in Figures 18 and 19 denote a unit area. As shown in Figure 18, dark-blue solid dots indicate that independent communication ranges of the G2 robots in structural types Fa and Ra are twelve and eighteen unit areas, respectively. The blue and light-blue solid dots represent the distinct communication domains of the G1 and G3 automata, encompassing sixteen and six units of area, correspondingly. Based on the above definitions, the independent communication ranges provided by six basic structural types are summarized in Table 8 (i.e., α_1 to α_6). However, if the left side of the focused G2 robot corresponds to the structural type Ra_1 , the focused G2 robot's two neighboring empty unit cells may be occupied by two *building* robots simultaneously (see Figure 19). In such cases, the focused G2 robot will go dormant and stop broadcasting, causing the reduction of the total communication area. According

to the *growing* robot's working modes, the *growing* robot that doesn't broadcast available empty unit cell coordinates is defined as the *dormant growing* sub-state (see Figure 2e). Although the overlapped communication area between neighboring *growing* robots can compensate for some of the losses, a *dormant* G2 robot can still lead to a reduction in the communication area by four unit areas (see Figure 19).

Table 8. The value of coefficients a_0 to a_6 .

α_0	α_1	α_2	α_3	α_4	α_5	α_6
4	18	36	24	42	12	18

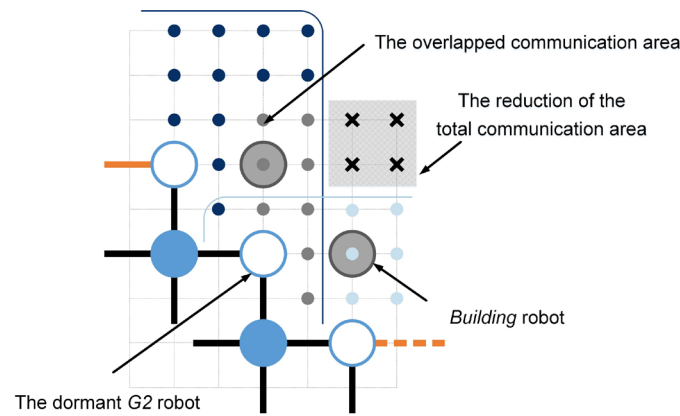


Figure 19. The effects of the *dormant* G2 robot for total communication area. Gray solid dots: the portion of the communication ranges overlap between a G2 robot and its neighboring *growing* robots. Black fork: The reduction of the total communication area.

Based on the definitions of the structural types, the calculation of the *dormant* G2 robots is determined by Equation (30).

$$D_{G2} = D_{G2}^{Ra_1} + D_{G2}^{Ra_2} + D_{G2}^{Fa} \quad (30)$$

where $D_{G2}^{Ra_1}$, $D_{G2}^{Ra_2}$, and D_{G2}^{Fa} denote the number of *dormant* G2 robots in structural types Ra_1 , Ra_2 , and Fa , respectively. The quantity of *dormant* G2 robots within structural types Ra_1 , i.e., $D_{G2}^{Ra_1}$, can be mathematically expressed as follows.

$$D_{G2}^{Ra_1} = N_{Ra_1}(t) \tau^* \mu_{Ra_1} \mu_{Ra_1} \quad (31)$$

where the value of τ^* is determined by Equation (27). The μ_{Ra_1} is the overall probability of the *building* robot being assigned to the building unit cell provided by the G2 robot of the structural type Ra_1 . It is represented in Equation (32).

$$\mu_{Ra_1} = \mu_{Ra_1}^{old} + \mu_{Ra_1}^{new} \quad (32)$$

here, $\mu_{Ra_1}^{old}$ and $\mu_{Ra_1}^{new}$ denote the overall probability that the building unit cells provided by the G2 robot of structural type Ra_1 are occupied by *old* and *new building* robots, respectively. According to the structural features analysis in Equations (19)–(25), $\mu_{Ra_1}^{old}$ and $\mu_{Ra_1}^{new}$ can be written as follows.

$$\mu_{Ra_1}^{old} = \frac{\xi \sigma_{old} + (N_{Ra_1}(t) - \xi) \epsilon_{old}}{N_{Ra_1}(t) + e_0} \quad (33)$$

$$\mu_{Ra_1}^{new} = \frac{\xi \sigma_{new} + (N_{Ra_1}(t) - \xi) \epsilon_{new}}{N_{Ra_1}(t) + e_0} \quad (34)$$

where e_0 can be found in Equation (21). Similarly, the number of *dormant* G2 robots within structural types Ra_2 and Fa , i.e., $D_{G2}^{Ra_2}$ and D_{G2}^{Fa} , can be written as follows, respectively.

$$D_{G2}^{Ra_2} = N_{Ra_2}(t)\tau\mu_{Ra_1}\mu_{Ra_2} \quad (35)$$

$$D_{G2}^{Fa} = N_{Fa}(t)\tau\mu_{Ra_1}\mu_{Fa} \quad (36)$$

where τ can be found in Equation (26). μ_{Ra_2} and μ_{Fa} represent the overall probability of the *building* robot distributing in the building unit cell provided by the G2 robot in structural types Ra_2 and Fa , respectively. They are written as follows.

$$\mu_{Ra_2} = \sigma_{new} + \sigma_{old} \quad (37)$$

$$\mu_{Fa} = \mu_{Fa}^{old} + \mu_{Fa}^{new} \quad (38)$$

here, σ_{old} and σ_{new} are shown in Equations (21) and (24). The μ_{Fa}^{old} and μ_{Fa}^{new} are the overall probability of *old building* and *new building* robots distributing in the building unit cells provided by the G2 robot in the structural type Fa . As analyzed in Equations (19)–(25), μ_{Fa}^{old} and μ_{Fa}^{new} can be expressed as

$$\mu_{Fa}^{old} = \frac{(N_{Fa_1*}(t) + N_{Fa_2*}(t))\sigma_{old} + (N_{Fa_1}(t) + N_{Fa_2}(t))\epsilon_{old}}{N_{Fa}(t) + e_0} \quad (39)$$

$$\mu_{Fa}^{new} = \frac{(N_{Fa_1*}(t) + N_{Fa_2*}(t))\sigma_{new} + (N_{Fa_1}(t) + N_{Fa_2}(t))\epsilon_{new}}{N_{Fa}(t) + e_0} \quad (40)$$

3.3.2. The State Transition Function $\Gamma_M^F(t)$

The $\Gamma_M^F(t)$ signifies the count of *moving* robots transitioning into *free* robots when they fail in competition with others for the same target position. This state transition function can be expressed as Equation (41).

$$\Gamma_M^F(t) = N_M(t)\gamma_f \quad (41)$$

where γ_f is the probability of the *moving* robots failing to compete for the target position. Indeed, γ_f is closely related to the working modes of the *growing* robots. Here, the *growing* robot that can provide an independent available empty unit cell for *moving* robots is regarded as being in the *working growing* sub-state, recorded as G_w (see Figure 2e). According to the proposed self-assembly strategy, some available empty unit cell coordinates only impact the *free* robots, not the *moving* ones. If the provided available empty unit cells are shared or do not affect the *moving* robots, the *growing* robot is considered to be in the *activated growing* sub-state (see Figure 2e). Each available empty unit cell affecting *moving* robots is shared by n_0 *moving* robots, i.e.,

$$n_0 = \frac{N_M(t)}{N_{G_w}(t)} \quad (42)$$

where $N_{G_w}(t)$ is the number of *working growing* robots. Therefore, the success and failure rates of *moving* robots competing to obtain a target position are γ_s and γ_f , i.e.,

$$\gamma_s = \frac{1}{n_0} \quad (43)$$

$$\gamma_f = 1 - \frac{1}{n_0} \quad (44)$$

The failure probability γ_f relies on the count of working *growing* robots $N_{G_w}(t)$.

According to the structural features, the calculation of the *working growing* state numbers for G1, G2, and G3 robots proceeds as follows. To determine G1 robots in the *working growing* state, two probabilistically independent events should be discussed. The first event is the probability of the G1 robot in *new growing* state. Here, the robot transformed from the *building* to the *growing* state at time step $t - 1$ is defined as the *new growing* robot at time step t , or it is regarded as the *old growing* robot (see Figure 2d). Following the proposed self-assembly strategy, the broadcasts of the *new growing* robots exclusively impact the *free* robots and not the *moving* robots. Thus, all *new growing* robot is in *activated growing* state. The probability of the G1 robot in a *new growing* state is expressed as ρ_{G1}^{new} , i.e.,

$$\rho_{G1}^{new} = \frac{\Lambda_B^{G1}(t-1, T_B)}{N_{G1}(t)} \quad (45)$$

here, Λ_B^{G1} is introduced in Section 3.3.4. The second event is the distribution characteristics of *building* robots within the building unit cells. Note that the possible building unit cell not occupied by a *building* robot will be preferentially selected as the available empty unit cell. Here, two situations should be considered. In the first situation, the building unit cell provided by a G1 robot is not occupied by a *building* robot. Note that this building unit cell must be shared by the neighboring G2 robot (see Section 3.2.3). Although the focused G1 robot and its neighboring G2 robot broadcast the building unit cell's coordinates simultaneously, only the G2 robot is usually considered the *working growing* state to avoid double counting. However, if the G2 robot is in a *new growing* state, the focused G1 robot could be considered the *working growing* robot. In addition, as G1 robot has three neighboring empty unit cells. Thus, when its building unit cell is occupied, the focused G1 robot will be in the *working growing* state. Thus, $N_{G1w}^F(t)$ is expressed as follows.

$$N_{G1w}^{Fa}(t) = [N_{Fa1*}(t) + N_{Fa2*}(t)](1 - \rho_{G1}^{new})[(1 - \sigma_{old} - \sigma_{new})\rho_{G2}^{new} + \sigma_{old}] \quad (46)$$

where $[N_{Fa1*}(t) + N_{Fa2*}(t)]$ expresses the total number of the G1 robot in structural types $Fa1^*$ and $Fa2^*$. The second terms on the right-hand side of Equation (46) indicate the discussed first events. Two sub-terms in the third term represent the discussed two situations in the second events, respectively. ρ_{G2}^{new} can be found in Equation (47).

Similarly, two probabilistically independent events need to be discussed to calculate the number of G2 robots in the *working growing* state. The first event is the probability of the G2 robots in a *new growing* state, i.e.,

$$\rho_{G2}^{new} = \frac{\Lambda_B^{G2}(t-1, T_B)}{N_{G2}(t)} \quad (47)$$

The second event is the structural features involving the arrangement characteristics of the structural types and the distribution characteristics of *building* robots within different building unit cells (See Section 3.2). Through a comprehensive analysis of the structural features, the G2 robot may be in the *working growing* state under the following three situations. The first situation is that the building unit cell provided by the focused G2 robot is not occupied by a *building* robot. Under this situation, the focused G2 robot is in the *working growing* state, providing an available empty unit cell for *free* and *moving* robots. The second situation is that an *old building* robot occupies the building unit cell provided by the focused G2 robot, and the left structural type of the focused G2 robot is not the structural type Ra_1 . Note that the G2 robot has two empty unit cell. According to the definition of the *building* robot's distribution, the focused G2 robot cannot actively use its second neighboring empty unit cell as a building unit cell (see Figure 14). Since the structural type on the left is not Ra_1 , the focused G2 robot can monopolize its second neighboring empty unit cell and work as a *working growing* robot to broadcast its second neighboring empty unit cell's coordinates. In the third situation, an *old building* robot occupies the focused G2 robot's building unit cell, and the left structural type is Ra_1 . As shown in Figure 16, the focused G2 robot's second

neighboring empty unit cell plays the role of the building unit cell of the G2 robot of the structural type Ra_1 . When the building unit cell of the G2 robot of the structural type Ra_1 is not occupied, the G2 robot of the structural type Ra_1 is prioritized as the *working growing* robot rather than the focused G2 robot to avoid double counting. However, if the G2 robot of the left structural type Ra_1 is in the *new growing* state and its building unit cell is not occupied, the focused G2 robot will be seen as a *working growing* robot. Additionally, a *new building* robot occupying a building unit cell indicates that the corresponding G2 robot completed a recruitment task at time step $t - 1$. Hence, the G2 robot cannot function as a *working growing* robot, as its broadcasts can only impact the *free* robots at time step t . Based on the above discussions, the number of G2 robots in structural type Ra_1 in the *working growing* state can be expressed as $N_{G2w}^{Ra_1}(t)$, i.e.,

$$N_{G2w}^{Ra_1}(t) = N_{Ra_1}(t)(1 - \rho_{G2}^{new}) \left[(1 - \mu_{Ra_1}) + \mu_{Ra_1}^{old}(1 - \tau^*) + \mu_{Ra_1}^{old}\tau^*(1 - \mu_{Ra_1})\rho_{G2}^{new} \right] \quad (48)$$

where $N_{Ra_1}(t)$ expresses the number of G2 robots in the structural type Ra_1 . Similarly, the number of G2 robots in structural types Ra_2 and Fa in the *working growing* state are written as $N_{G2w}^{Ra_2}(t)$ and $N_{G2w}^{Fa}(t)$, i.e.,

$$N_{G2w}^{Ra_2}(t) = N_{Ra_2}(t)(1 - \rho_{G2}^{new}) \left[(1 - \mu_{Ra_2}) + \sigma_{old}(1 - \tau) + \sigma_{old}\tau(1 - \mu_{Ra_1})\rho_{G2}^{new} \right] \quad (49)$$

$$N_{G2w}^{Fa}(t) = N_{Fa}(t)(1 - \rho_{G2}^{new}) \left[(1 - \mu_{Fa}) + \mu_{Fa}^{old}(1 - \tau) + \mu_{Fa}^{old}\tau(1 - \mu_{Ra_1})\rho_{G2}^{new} \right] \quad (50)$$

As introduced in Section 3.2.3, the G3 robots cannot enter a *new growing* state, and their only neighboring empty unit cells are the priority building unit cells by default. In the structural types Fa_2 and Fa_{2*} , the G3 robot can use its building unit cell preferentially. Thus, as long as a *building* robot does not occupy the building unit cell, the G3 robot in the structural types Fa_2 and Fa_{2*} will remain in the *working growing* state. Based on the structural features, the quantity is expressed as $N_{G3w}^{Fa}(t)$, i.e.,

$$N_{G3w}^{Fa}(t) = [N_{Fa_2}(t) + N_{Fa_{2*}}(t)](1 - \sigma_{old} - \sigma_{new}) \quad (51)$$

In the structural type Ra_2 , G2 and G3 robots share the building unit cell. When the *building* robot does not occupy the building unit cell, the G2 robot is prioritized as the *working growing* robot to avoid double counting. However, if the G2 robot is in the *new growing* state, the G3 robot may be in the *working growing* state. It is expressed as follows.

$$N_{G3w}^{Ra_2}(t) = N_{Ra_2}(t)(1 - \sigma_{old} - \sigma_{new})\rho_{G2}^{new} \quad (52)$$

Finally, the total number of the *working growing* robots is written as follows.

$$N_{Gw}(t) = N_{G2w}^{Ra_1}(t) + N_{G2w}^{Ra_2}(t) + N_{G2w}^{Fa}(t) + N_{G3w}^{Fa}(t) + N_{G3w}^{Ra_2}(t) + N_{G1w}^{Fa}(t) \quad (53)$$

3.3.3. The State Transition Functions $\Gamma_M^B(t)$ and $\Lambda_M^F(t, T_M)$

According to the self-assembly strategy, all failed *moving* robots in competition will actively degrade to the *free* state. As shown in Equations (42) and (43), the number of *moving* robots winning the competition equals to the number of available empty unit cells, i.e., $N_{Gw}(t)$. Based on **Assumption 1**, the probability of these winning *moving* robots transforming into *building* robots at time step t can be regarded as a constant-coefficient φ . Indeed, like the ξ in Equation (20), the value of the constant-coefficient φ is also closely related to the autonomous actions of robots and is affected by the building time parameter T_B . Thus, based on lots of simulation results, the value of φ under different building time

parameters T_B is shown in Table 7. Then, the state transition function from *moving* robots to *building* robots is written as in Equation (54), i.e.,

$$\Gamma_M^B(t) = \varphi N_{G_w}(t) \quad (54)$$

The $\Lambda_M^F(t, T_M)$ represents the number of *moving* robots transitioning into *free* robots due to running out of the moving time parameter T_M . It is equivalent to the number of *moving* robots that have not changed their state during the time interval $[t - T_M, t]$. Referring to Ref. [16], $\Lambda_M^F(t, T_M)$ can be expressed as follows.

$$\Lambda_M^F(t, T_M) = \Gamma_F(t - \tau_m) \prod_{k=t-\tau_m+1}^t (1 - \gamma_f - r_M^B(t)) \quad (55)$$

where $\Gamma_M(t - \tau_m)$ represents the number of *moving* robots transformed from the *free* robots at time step $t - \tau_m$. γ_f can be found in Equation (44). The $r_M^B(t)$ is the probability of *moving* robots transforming into a *building* state at time step t . It is expressed as follows.

$$r_M^B(t) = \frac{\Gamma_M^B(t)}{N_M(t)} \quad (56)$$

3.3.4. The State Transition Functions $\Lambda_B^G(t, T_B)$, $\Lambda_B^{G1}(t, T_B)$, and $\Lambda_B^{G2}(t, \tau_B)$

The state transition function $\Lambda_B^G(t, T_B)$ represents the number of robots transitioning from *building* state to *growing* state at time step t . As introduced in Section 2.2, all *building* robots generated at the $t - T_B$ will become the *growing* robots at time step t . Thus, the $\Lambda_B^G(t, T_B)$ is written as follows.

$$\Lambda_B^G(t, T_B) = \Gamma_M^B(t - T_B) \quad (57)$$

As analyzed in Section 3.2.3, the *old building* robots are preferentially distributed in the priority building unit cells. Thus, the robots transitioning from *building* to *growing* state will be preferentially distributed in the priority building unit cells. Its probability can be written as follows.

$$\sigma_{\Lambda_B} = \frac{\Lambda_B^G(t, T_B)}{N_{bu_p}(t) + e_0} \quad (58)$$

Correspondingly, the probability of these robots being distributed in the non-priority building unit cells is as follows.

$$\epsilon_{\Lambda_B} = \frac{\Lambda_B^G(t, T_B) - \sigma_{\Lambda_B} N_{bu_p}(t)}{N_{bu}(t) - N_{bu_p}(t) + e_0} \quad (59)$$

As shown in Figure 14, only the structural types Fa_1 and Fa_2 can create $G1$ robots. Thus, the number of robots transforming from *building* to $G1$ state can be written as follows.

$$\Lambda_B^{G1}(t, T_B) = N_{Fa_1}(t) \epsilon_{\Lambda_B} + N_{Fa_2}(t) [\sigma_{\Lambda_B} (1 - \epsilon_{\Lambda_B}) + \epsilon_{\Lambda_B} (1 - \sigma_{\Lambda_B})] \quad (60)$$

Similarly, the $G2$ robots can born in the structural types Ra_1 , Ra_2 , Fa_{1*} , and Fa_{2*} . Note that the structural type Fa_2 provides two adjacent building unit cells. Thus, when two *building* robots simultaneously occupy these two adjacent building unit cells, they will become the $G2$ sub-state together. The number of robots transforming from *building* to $G2$ state can be written as follows.

$$\Lambda_B^{G2}(t, \tau_B) = \sigma_{\Lambda_B} (N_{Ra_2}(t) + N_{Fa_{1*}}(t) + 2N_{Fa_{2*}}(t) + \xi) + \epsilon_{\Lambda_B} (N_{Ra_1}(t) - \xi) + 2N_{Fa_2}(t) \sigma_{\Lambda_B} \epsilon_{\Lambda_B} \quad (61)$$

3.3.5. The State Transition Functions $\Delta_{G1}^{G2}(t)$, $\Delta_{G1}^{G3}(t)$, $\Delta_{G2}^{G3}(t)$, and $\Delta_{G3}^S(t)$

As *building* robots transform into a *growing* state, the existing *growing* robots will upgrade their sub-states or transform into the *solid* state. In the structural types Fa_{1*} and Fa_{2*} , creating a *growing* robot will make the $G1$ robot update to the $G2$ robot. Thus, the number of $G1$ robots updating to the $G2$ sub-state can be expressed in Equation (62) based on the analysis in Equations (58) and (59).

$$\Delta_{G1}^{G2}(t) = \sigma_{\Lambda_B} N_{Fa_{1*}}(t) + 2\sigma_{\Lambda_B}(1 - \sigma_{\Lambda_B})N_{Fa_{2*}}(t) - \Omega_{G1}^*(t) \quad (62)$$

here, $\Omega_{G1}^*(t)$ represents the number of $G1$ robots that trigger the constraints of the shape boundary and become the *solid* robot directly during the update of the sub-state. The $G1$ robots described by $\Omega_{G1}^*(t)$ belong to the *old growing* robots. However, it is also possible that a *growing* robot triggering shape boundary constraints belongs to the *new growing* robot. The new and old states of these *growing* robots depend on the specified 2D shape and the stochastic factors in the self-assembly process. Thus, for generality and simplicity, the $\Omega_{G1}^*(t)$ is defaulted to half of $\Omega_{G1}(t)$, i.e.,

$$\Omega_{G1}^*(t) = 0.5\Omega_{G1}(t) \quad (63)$$

where $\Omega_{G1}(t)$ can be found in Equation (72). As shown in Figure 14f, when two *growing* robots are born in these two building unit cells simultaneously, the $G1$ robot in the structural type Fa_{2*} will directly update to the $G3$ state. Thus, $\Delta_{G1}^{G3}(t)$ can be written as follows.

$$\Delta_{G1}^{G3}(t) = \sigma_{\Lambda_B}^2 N_{S_{F2*}}(t) \quad (64)$$

Each structural type has a $G2$ robot. Thus, the number of $G2$ robots transforming into $G3$ robots can be expressed as follows.

$$\Delta_{G2}^{G3}(t) = \sigma_{\Lambda_B} (N_{Fa_{1*}}(t) + N_{Fa_{2*}}(t) + N_{Ra_2}(t) + 2\zeta) + \epsilon_{\Lambda_B} (N_{Fa_1}(t) + N_{Fa_2}(t) + 2(N_{Fa_1}(t) - \zeta)) - \Omega_{G2}^*(t) \quad (65)$$

Here, $\Omega_{G2}^*(t)$ represents the number of *old* $G2$ robots that trigger the constraints of the shape boundary and become the *solid* robot directly during the update of the sub-state. Referring to Equation (63), it can be expressed as follows.

$$\Omega_{G2}^*(t) = 0.5\Omega_{G2}(t) \quad (66)$$

where $\Omega_{G2}(t)$ can be calculated in Equation (71). $G3$ robots exist in the structural types Fa_2 , Fa_{2*} , and Ra_2 . Thus, the number of $G3$ robots transforming into *solid* robots can be expressed as follows.

$$\Delta_{G3}^S(t) = \sigma_{\Lambda_B} (N_{Fa_2}(t) + N_{Fa_{2*}}(t) + N_{Ra_2}(t)) \quad (67)$$

Note that $G3$ robot will not trigger the shape boundary constraints.

3.4. The Constraints of the Shape Boundary

The shape of the formation is the collective goal of the entire swarm robotics system, specified and designed by the user. However, the shape boundary can partition the cross-arranged unit cells, causing it to be incomplete. Thus, when situated along the shape boundary, $G1$ and $G2$ robots transition directly to the *solid* state due to this incomplete unit cell. The predefined shape allows us to quantify how the shape boundary constraints affect the state transitions of the *growing* robots. The calculation results can serve as the initial conditions for the macroscopic model.

Using the example of a four-pointed star shape formation comprising 285 robots, 68 robots are affected by the shape boundary (see Figure 20b). To simplify our analysis, the robots containing two or three neighboring filled unit cells are considered the $G2$

robots triggering the shape boundary constraints. Others are considered the *G1* robots triggering the shape boundary constraints. Thus, as shown the blue and red hollow circles in Figure 20b, 60 *G2* and 8 *G1* robots trigger the shape boundary constraints. Additionally, the number of *growing* robots affected by the shape boundary at time step t is assumed to increase linearly as the total number of *growing* and *solid* robots increases. Here $x(t)$ is used to express the total number of *growing* and *solid* robots in the formed formation at time step t . It is written as follows.

$$x(t) = N_G(t) + N_S(t) \quad (68)$$

The following equations can express the relationship between the number of *G2* robots affected by the boundary and the total number of *growing* and *solid* robots x .

$$\int_{x_{G2}^0}^{x_{G2}^1} k_{G2}x + b_{G2} dx = n_t \quad (69)$$

$$k_{G2}x_{G2}^0 + b_{G2} = 0 \quad (70)$$

where n_t is the total number of *G2* robots affected by the shape boundary. In this case, n_t is equal to 60. x_{G2}^0 is the maximum number of *growing* and *solid* robots that can be accommodated in the formed formation before the first *G2* robot triggers the boundary constraints (see Figure 20a). After the last *G2* robot triggers the boundary constraints, the number of robots in the formed formation is represented by x_{G2}^1 (see Figure 20b). Here, the x_{G2}^0 and x_{G2}^1 are 137 and 277, respectively. By simultaneously solving Equations (69) and (70), we obtain $k_{G2} = 3/490$, and $b_{G2} = -411/490$. Therefore, the number of *G2* robots affected by constraints of the shape boundary at time step t can be expressed as follows.

$$\Omega_{G2}(t) = \int_{N_G(t-1)+N_S(t-1)}^{N_G(t)+N_S(t)} k_{G2}x + b_{G2} dx, \quad x_{G2}^1 \geq x > x_{G2}^0 \quad (71)$$

Similarly, the number of *G1* robots affected by constraints of the shape boundary at time step t can be expressed as follows.

$$\Omega_{G1}(t) = \int_{N_G(t-1)+N_S(t-1)}^{N_G(t)+N_S(t)} k_{G1}x + b_{G1} dx, \quad x_{G1}^1 \geq x > x_{G1}^0 \quad (72)$$

Taking Figure 20b as an example, the k_{G1} , b_{G1} , x_{G1}^0 , and x_{G1}^1 are equal to 0.25, -69.25 , 277, and 285.

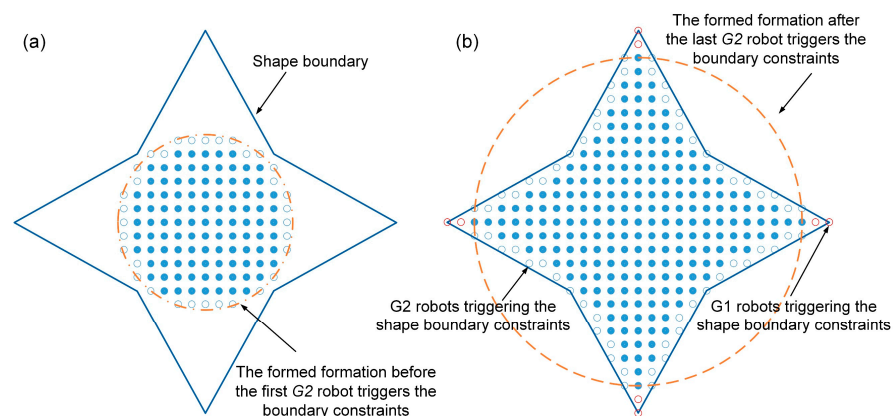


Figure 20. The calculating example of the constraints of the shape boundary. (a) The initial condition triggers the constraints of the shape boundary. (b) The final condition triggers the constraints of the shape boundary.

4. Verification and Discussion

4.1. The Verification of the Macroscopic Model

The prediction performance is the key indicator used to verify the proposed macroscopic model. In this section, we introduce the prediction error index (P_{EI}) to assess the model's prediction accuracy. The P_{EI} considers the prediction accuracy of *moving*, *building*, *growing*, and *solid* states. Based on the Euclidean distance and relative error, the prediction error index is defined as follows.

$$P_{EI} = \frac{1}{T} \sum_{t=1}^T \frac{|N_M^M(t) - N_M^E(t)| + |N_B^M(t) - N_B^E(t)| + |N_G^M(t) - N_G^E(t)| + |N_S^M(t) - N_S^E(t)|}{n^E(t)} \quad (73)$$

where T is the total time of the self-assembly formation. N_M^M , N_B^M , N_G^M , and N_S^M are the predicted number of *moving*, *building*, *growing*, and *solid* robots generated by the proposed macroscopic model. N_M^E , N_B^E , N_G^E , and N_S^E are the average results of *moving*, *building*, *growing*, and *solid* robots in multiple experiments. The experimental results may come from simulations or real-robot experiments. $n^E(t)$ can be expressed as follows.

$$n^E(t) = N_M^E(t) + N_B^E(t) + N_G^E(t) + N_S^E(t) \quad (74)$$

In this section, the four-pointed star shape formation is selected to verify the prediction performance of the proposed macroscopic model. The discussion of prediction performance in the small-scale self-assembly formation uses the scenario shown in Figure 10a. Here, 15 robots are used to self-assembly formation. The values of building time parameter T_B and moving time parameter T_M are set as one step and four steps, respectively. Here, both the real-robot experiments based on Waxberry robots and simulations based on embodied simulators are carried out. The real-robot experiment is repeated 10 times, and the simulations is repeated 100 times. With the same system parameters and specified scale, the prediction results of the whole self-assembly formation process are also calculated based on the proposed macroscopic model. As shown in Figure 21, the predictions based on the macroscopic model, the average results of 100 simulations and the average results of 10 real-robot experiments are represented by blue solid line, red dotted line, and green solid line. The black line with dots represents the experiment result in Figure 10a. Figure 21 indicates that the proposed macroscopic model can accurately describe each state's change process. Compared with the average results of 10 real-robot experiments and the average results of 100 simulations, the proposed model's prediction error index for this small-scale self-assembly formation is 14.17% and 12.20%, respectively, according to Equation (73). Note that the predicted result of the macroscopic model is not a reproduction of a single experiment. Instead, it directly feeds back the average results obtained in multiple experiments under specified scenarios. Although some errors exist between the experiment results of Figure 10a and the model's predicted results, their dynamic trends are consistent.

Indeed, when the size of the specified shape formation is very small, the self-assembly formation process is easily affected by random factors. The characteristics of the prediction performance of the proposed model cannot be displayed clearly. Therefore, a large-scale formation consisting of 300 robots is also used to verify the proposed model. The building time parameter T_B and moving time parameter T_M are set as three steps and four steps, respectively. After 100 simulations, the comparison between the simulation's average results and the model's predictions is shown in Figure 22. The proposed model's prediction error index for this large-scale self-assembly formation is only 4.05% according to Equation (73). As shown in Figure 22, the difference between the prediction and simulation results mainly occurs in the second half of the self-assembly formation process. This is caused by the constraints of the shape boundary being triggered. Indeed, the constraints of the shape boundary may create other complex structural types not considered in Figure 14. Meanwhile, the impact of boundary constraints on the quantities of six basic structural types is also ignored in Section 3.2.1. Therefore, the accuracy of the model's prediction results

will decrease after triggering the boundary constraints. When predicting the number of *G1* robots, the error is particularly evident (see Figure 22f). However, since the number of *G1* robots is generally very small, this does not significantly increase in the prediction error index P_{EI} .

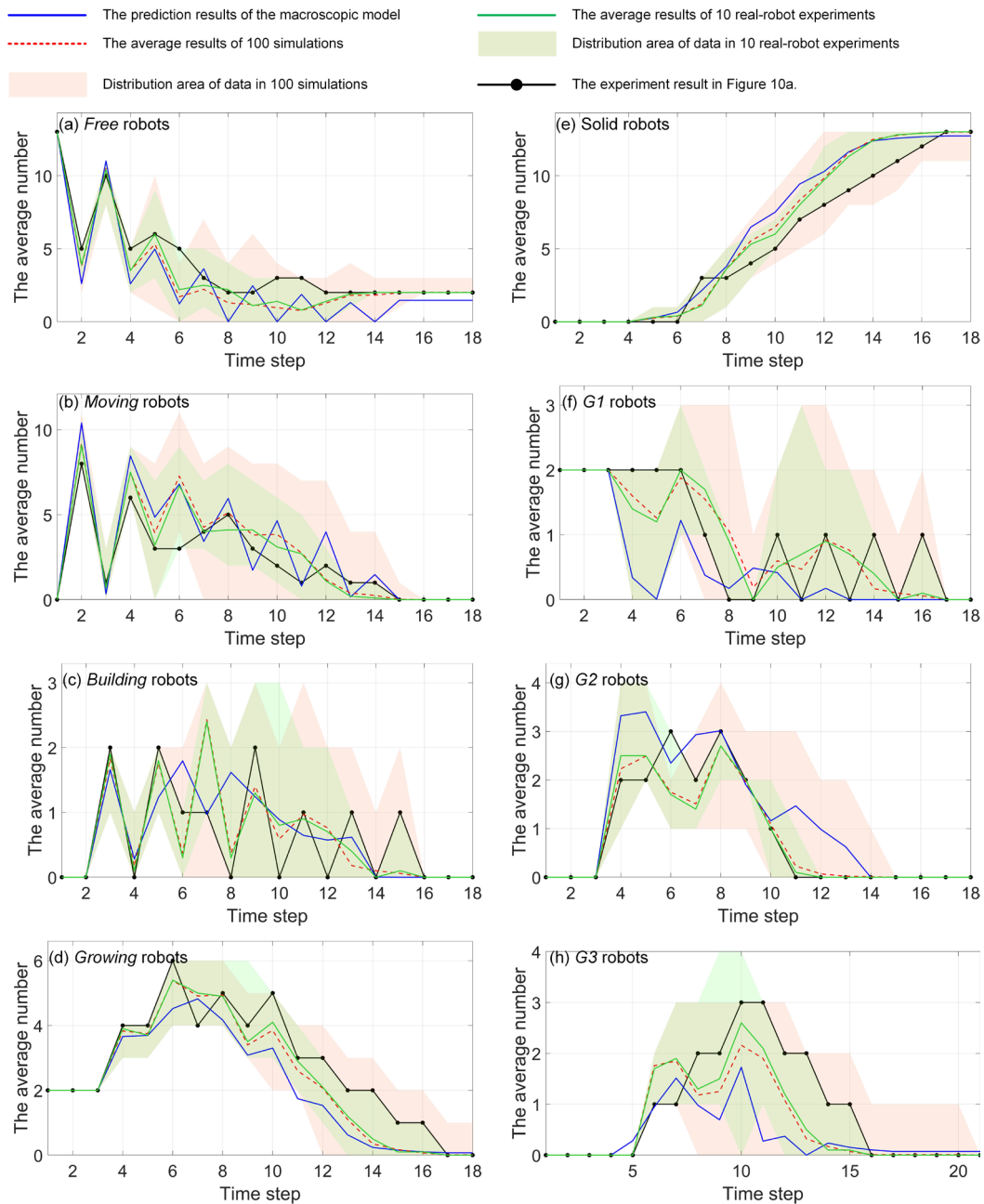


Figure 21. The analysis of the prediction performance of the macroscopic model in a small-scale four-pointed star shape formation. Blue solid line: the prediction results of the macroscopic model. Red dotted line: the average results of 100 simulations. Red shadow: distribution area of data in 100 simulations. Green solid line: the average results of 10 real-robot experiments. Green shadow: distribution area of data in 10 real-robot experiments. The black line with dots: the experiment result in Figure 10a.

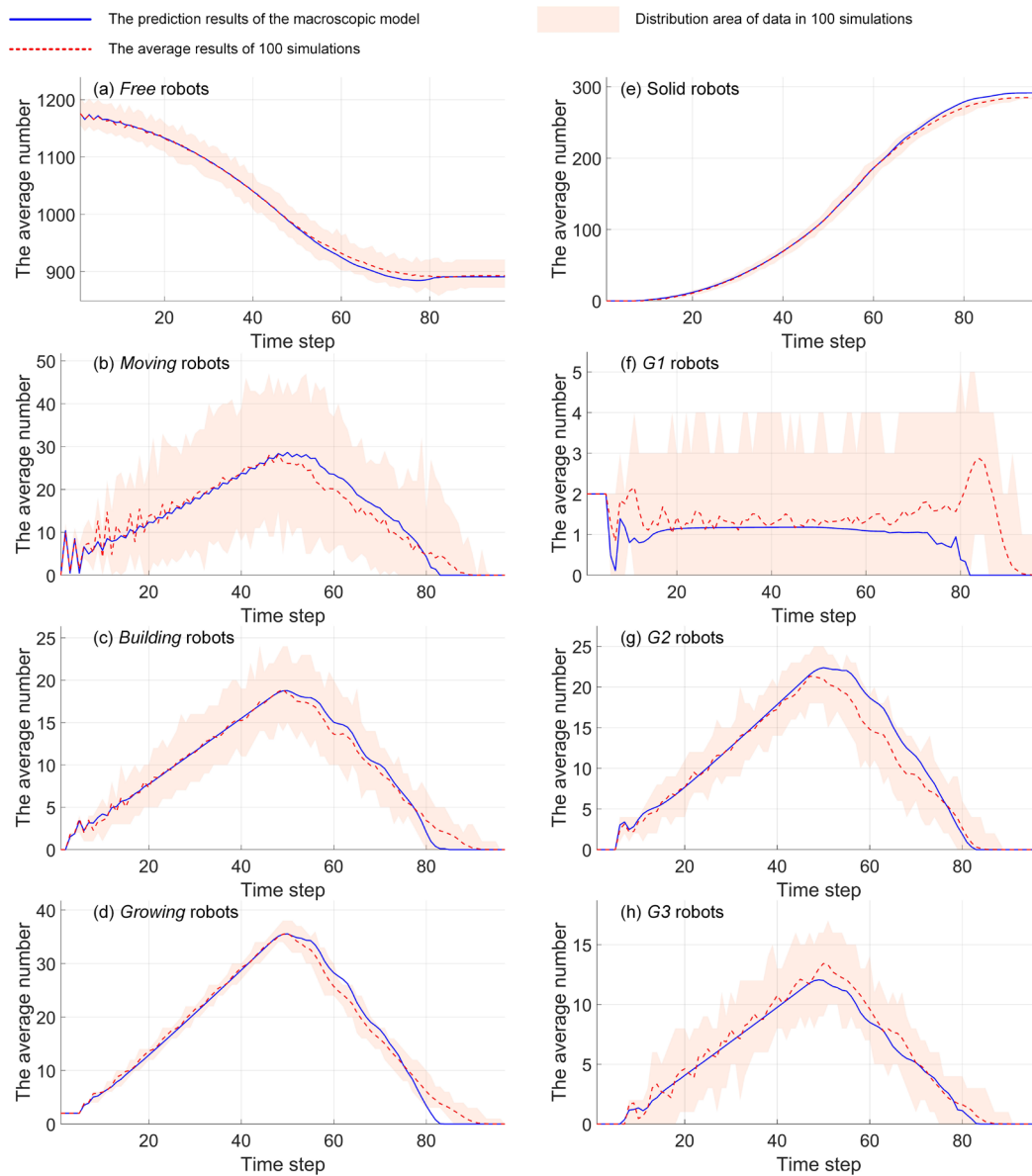


Figure 22. The analysis of the prediction performance of the macroscopic model in a large-scale four-pointed star shape formation. Blue solid line: the prediction results of the macroscopic model. Red dotted line: the average results of 100 simulations. Red shadow: distribution area of data in 100 simulations.

To assess the adaptability of the proposed macroscopic model, the prediction error indexes are tested across various parameters using the four-pointed star and hexagonal crystal as formation shapes. The formation scale involves approximately 15 robots, 300 robots, 1500 robots, and 3500 robots, with the building time parameter T_B set from 1 to 6 steps. Based on the analysis in Figure 21, the prediction error index of less than 20% is considered credible. As shown in Figure 23, the prediction error indexes of the proposed model show a similar trend in the two shapes. The prediction error indexes decrease with the scale of the formation in both shapes. These observations suggest that the proposed model demonstrates high adaptability across different shapes and different formation scales. Additionally, the proposed model proves effective in predicting the change process of self-assembly (i.e., $P_{EI} < 20\%$) when the building time parameter T_B is less than five steps. Notably, when the T_B is set to three steps, the prediction error index reaches its lowest point. Especially when the formation scale is up to about 3500 robots, the P_{EI} is less than 4% in both shapes. However, the model exhibits poor applicability

when the building time parameter exceeds four steps. This limitation can be attributed to the distribution characteristics of *building* robots discussed in Section 3.2.3. As shown in Figure 14, the proposed model considers only part of neighboring empty unit cells as the building unit cells for *building* robots. Yet, with an increase in the building time parameter, all neighboring empty unit cells have the potential to become building unit cells and be occupied by *building* robots. Unfortunately, such scenarios are not accounted for in the proposed model, leading to a decrease in its prediction performance. The preceding analysis highlights that the proposed model shows excellent adaptability in predicting the self-assembly formation across diverse shapes, scales, and system parameters.

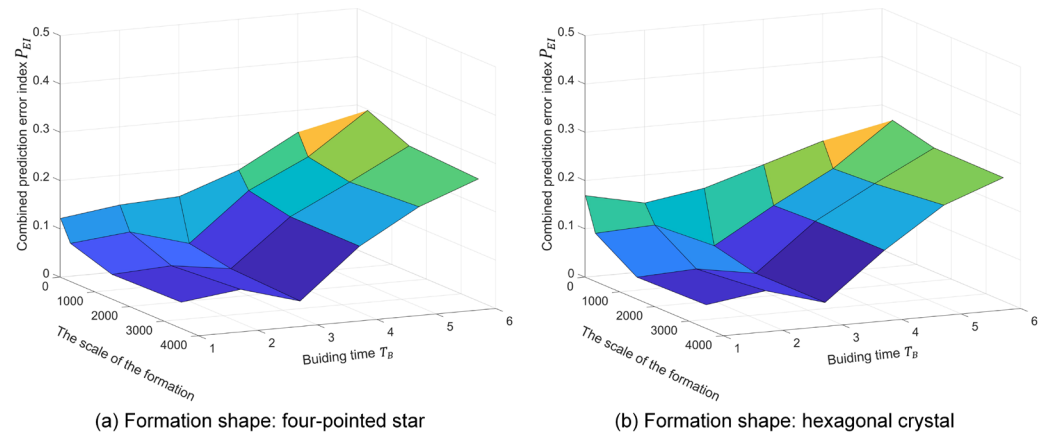


Figure 23. The prediction error indexes of the model in different conditions.

4.2. The Discussions

Compared to simulation and real-robot experiments, the macroscopic models can more efficiently display swarm behavior under varying influence factors. In this Section, we will discuss what factors affect the forming efficiency of the proposed self-assembly formation by using the four-pointed star shape formation consisting of about 300 robots as an example. Based on the proposed macroscopic model, the impact of the system parameters, i.e., building time parameter T_B and moving time parameter T_M , should be first discussed. As shown in Figure 24, the total time of the self-assembly formation will increase with the building time parameter T_B , regardless of the formation's scale. It is evident that the larger the formation scale, the more significant the impact of the building time parameter on the forming efficiency. In the formation consisting of about 3500 robots, reducing the building time parameter from 4 steps to 1 step resulted in a remarkable 28.5% reduction in the time required for formation. In the formation consisting of about 15 robots, a 16% reduction in total self-assembly time is observed when the building time parameter decreases from 4 steps to 1 step.

The moving time parameter T_M is used to avoid the downtime of the self-assembly due to the fault of the *moving* robots. The effect of the moving time parameter on forming efficiency is another concern, as more *moving* robots will be forced to degenerate into *free* robots by reducing the moving time parameter. To analyze the impact of the moving time parameter on forming efficiency, the proposed model calculates the self-assembly process for moving time parameters from one step to four steps, respectively. The building time parameter is defaulted to three steps. Note that the state transition function from *moving* robots to *building* robots at time step t , i.e., $\Gamma_M^B(t)$, determines the forming efficiency. Thus, $\Gamma_M^B(t)$ is selected to analyze the impact of the moving time parameter on forming efficiency. As shown in Figure 25, the forming efficiency of the self-assembly formation is almost constant, regardless of the value of the moving time parameter. Especially when the moving time parameter is more than four steps, these curves of the state transition function $\Gamma_M^B(t)$ values almost overlap. That is, the moving time parameter T_M does not affect the formation efficiency of the self-assembly formation. The reason for this counterintuitive

result may come from the design of the proposed self-assembly formation strategy. As introduced in Section 2.2, each target position is shared by multiple *moving* robots, meaning high redundancy. Although some *moving* robots are limited by the moving time parameter being forced to degenerate into *free* robots, their target positions will still be finished by other *moving* robots.

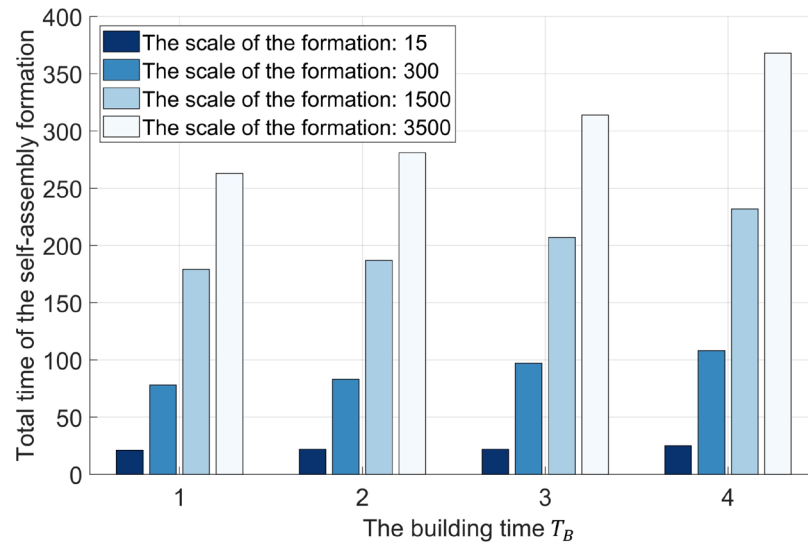


Figure 24. The effects of the building time parameter T_B on the forming efficiency.

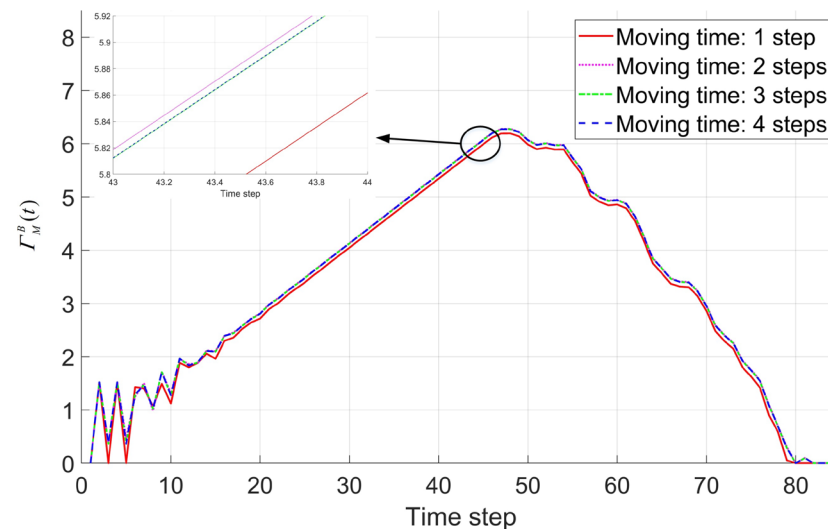


Figure 25. The effects of the moving time parameter T_M on the forming efficiency.

The simulation tests for the parameters T_B and T_M prove that the formation efficiency of the proposed self-assembly formation exhibits monotonicity on the building time parameter T_B and is independent of the moving time parameter T_M . Indeed, the building time parameter T_B generally depends on the performance of a single robots. The faster the *building* robot finishes its assembly work, the less time there is for the self-assembly formation. However, the moving time parameter T_M has nothing to do with the performance of a single robot. These results illustrate that the proposed distributed self-assembly formation strategy fully utilizes the performance of individual robots. The higher the individual performance (i.e., reducing building time), the higher the forming efficiency of the proposed self-assembly formation. This points out ways of optimizing the efficiency of the self-assembly formation. Meanwhile, the analysis of moving time parameters T_M

indicates that the proposed self-assembly formation has strong self-stability. Parameters unrelated to individual performance do not affect the swarm's forming efficiency.

5. Conclusions

In this paper, a high-performance distributed self-assembly formation strategy is proposed, inspired by crystallization. To convert the macroscopic swarm behaviors to multiple simple collaborative tasks, five robot states, i.e., *free*, *moving*, *building*, *growing*, and *solid*, are defined by imitating crystallization. Consequently, robots in different states spontaneously formed a specified 2D shape formation with a lattice structure through local interactions. The feasibility of the proposed strategy is verified using 15 Waxberry robots to form the four-pointed star shape and hexagonal crystal shape. Further self-assembly formation tests in different conditions and larger scales were also conducted, facilitated by an embodied simulator. The fully distributed strategy and adaptation in different formation shapes verify the flexibility of the proposed self-assembly formation. The scalability of the strategy is confirmed through tests encompassing formations ranging from approximately 15 robots to 3500 robots. In addition, since all *growing* robots contribute to the expansion of the formation, the proposed self-assembly formation exhibits high forming efficiency. A rare and precious super-linear trend further demonstrates the high forming efficiency. Specifically, as the formation scale increases from about 15 to 3500 robots, the efficiencies of forming the four-pointed star shape and the hexagonal crystal shape formations are increased by 1756% and 2417%, respectively. These results collectively affirm the flexibility, scalability, and high efficiency of the proposed self-assembly formation strategy. The proposed high-performance self-assembly strategy for swarm robotics holds significant implications for advancing the practical applications of swarm robotics. For example, in flood relief efforts, many aquatic robots can rapidly self-assemble into emergency floating bridges based on the proposed self-assembly strategy. In intelligent warehouse systems, multiple transport robots can self-assemble into different-sized transport equipment using the proposed self-assembly strategy, enabling automatic adaptation to transport items of varying sizes.

In addition, a non-spatial macroscopic model is further developed to predict and analyze the swarm behavior. Following the stock and flow diagram generated by FSM, the master equations of this model are created. To solve the challenges of constructing the model's state transition functions caused by the complex spatial structure, the SFE method is first proposed. This method opens the door to modeling the model for the swarm behavior affected by complex spatial structure. Here, we define six basic structural types and discuss their characteristics in detail. On this basis, all state transition functions that can explain the transition mechanisms are constructed. The simulations and experiments show that the proposed model has excellent prediction performance (i.e., $P_{EI} < 20\%$) when the value of the building time parameter T_B is less than five steps. The analysis based on the model demonstrates that it fully utilizes the performance of individual robots. A shorter building time consumed by a single robot means a higher forming efficiency. Additionally, the analysis of the moving time parameter T_M verifies that the proposed self-assembly formation has strong self-stability. Parameters independent of individual performance do not affect the swarm's forming efficiency.

Due to the lack of more Waxberry robots, the self-assembly formation has not been tested on larger-scale real robots. Producing more Waxberry robots and completing the test with larger-scale real robots is a focus of future research. Furthermore, although the proposed macroscopic model can adapt to parameter changes within a certain range, this adaptability is restricted to regions where the building time parameter T_B is less than five steps. Therefore, further improving the model so that it can be adapted to all scenarios is another important research topic for the future.

Author Contributions: Conceptualization, Z.P. and G.W.; methodology, Z.P. and S.Y.; software, Z.P. and Z.T.; writing—original draft preparation, Z.P.; writing—review and editing, G.W., H.Y. and S.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Nos. 11832009, 12172095, 11902085), the Full-time Introduction of National High-level Innovation Talents Research Project of Hebei Province (No. 2021HBQZYCSB003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Nasseri, M.A.; Asadpour, M. Control of flocking behavior using informed agents: An experimental study. In Proceedings of the 2011 IEEE Symposium on Swarm Intelligence, Paris, France, 11–15 April 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 1–6.
- Zhang, Z.; Yang, T.; Zhang, T.; Zhou, F.; Cen, N.; Li, T.; Xie, G. Global vision-based formation control of soft robotic fish swarm. *Soft Robot.* **2021**, *8*, 310–318. [\[CrossRef\]](#) [\[PubMed\]](#)
- Spears, W.M.; Spears, D.F.; Hamann, J.C.; Heil, R. Distributed, physics-based control of swarms of vehicles. *Auton. Robot.* **2004**, *17*, 137–162. [\[CrossRef\]](#)
- Hamann, H. *Swarm Robotics: A Formal Approach*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 221.
- Tan, Y.; Zheng, Z. Research advance in swarm robotics. *Def. Technol.* **2013**, *9*, 18–39. [\[CrossRef\]](#)
- Dias, P.G.F.; Silva, M.C.; Rocha Filho, G.P.; Vargas, P.A.; Cota, L.P.; Pessin, G. Swarm robotics: A perspective on the latest reviewed concepts and applications. *Sensors* **2021**, *21*, 2062. [\[CrossRef\]](#) [\[PubMed\]](#)
- Brambilla, M.; Ferrante, E.; Birattari, M.; Dorigo, M. Swarm robotics: A review from the swarm engineering perspective. *Swarm Intell.* **2013**, *7*, 1–41. [\[CrossRef\]](#)
- Dorigo, M.; Theraulaz, G.; Trianni, V. Swarm robotics: Past, present, and future [point of view]. *Proc. IEEE* **2021**, *109*, 1152–1165. [\[CrossRef\]](#)
- Firat, Z.; Ferrante, E.; Gillet, Y.; Tuci, E. On self-organised aggregation dynamics in swarms of robots with informed robots. *Neural Comput. Appl.* **2020**, *32*, 13825–13841. [\[CrossRef\]](#)
- Song, Y.; Fang, X.; Liu, B.; Li, C.; Li, Y.; Yang, S.X. A novel foraging algorithm for swarm robotics based on virtual pheromones and neural network. *Appl. Soft Comput.* **2020**, *90*, 106156. [\[CrossRef\]](#)
- Hinsen, P.; Wiedemann, T.; Shutin, D.; Lilienthal, A.J. Exploration and Gas Source Localization in Advection–Diffusion Processes with Potential-Field-Controlled Robotic Swarms. *Sensors* **2023**, *23*, 9232. [\[CrossRef\]](#)
- Cavalcanti, A.; Miyazawa, A.; Sampaio, A.; Li, W.; Ribeiro, P.; Timmis, J. *Modelling and Verification for Swarm Robotics*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 1–19.
- Ijspeert, A.J.; Martinoli, A.; Billard, A.; Gambardella, L.M. Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. *Auton. Robot.* **2001**, *11*, 149–171. [\[CrossRef\]](#)
- Martinoli, A.; Easton, K.; Agassounon, W. Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *Int. J. Robot. Res.* **2004**, *23*, 415–436. [\[CrossRef\]](#)
- Lerman, K.; Galstyan, A.; Martinoli, A.; Ijspeert, A. A macroscopic analytical model of collaboration in distributed robotic systems. *Artif. Life* **2001**, *7*, 375–393. [\[CrossRef\]](#) [\[PubMed\]](#)
- Liu, W.; Winfield, A.F.; Sa, J. Modelling swarm robotic systems: A case study in collective foraging. *Towards Auton. Robot. Syst.* **2007**, *23*, 25–32.
- Whitesides, G.M.; Grzybowski, B. Self-assembly at all scales. *Science* **2002**, *295*, 2418–2421. [\[CrossRef\]](#)
- Groß, R.; Bonani, M.; Mondada, F.; Dorigo, M. Autonomous self-assembly in swarm-bots. *IEEE Trans. Robot.* **2006**, *22*, 1115–1130.
- Yang, J.; Wang, X.; Bauer, P. Line and v-shape formation based distributed processing for robotic swarms. *Sensors* **2018**, *18*, 2543. [\[CrossRef\]](#) [\[PubMed\]](#)
- Ayre, M.; Izzo, D.; Pettazzi, L. Self assembly in space using behaviour based intelligent components. *Taros Towards Auton. Robot. Syst.* **2005**.
- Sabatini, M.; Reali, F.; Palmerini, G.B. Autonomous behavioral strategy and optimal centralized guidance for on-orbit self assembly. In Proceedings of the 2009 IEEE Aerospace Conference, Big Sky, MT, USA, 7–14 March 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 1–12.
- McMullen, A.; Muñoz Basagoiti, M.; Zeravcic, Z.; Brujic, J. Self-assembly of emulsion droplets through programmable folding. *Nature* **2022**, *610*, 502–506. [\[CrossRef\]](#) [\[PubMed\]](#)
- Goldstein, S.C.; Campbell, J.D.; Mowry, T.C. Programmable matter. *Computer* **2005**, *38*, 99–101. [\[CrossRef\]](#)
- Douglas, S.M.; Dietz, H.; Liedl, T.; Högberg, B.; Graf, F.; Shih, W.M. Self-assembly of DNA into nanoscale three-dimensional shapes. *Nature* **2009**, *459*, 414–418. [\[CrossRef\]](#)
- Rubenstein, M.; Cornejo, A.; Nagpal, R. Programmable self-assembly in a thousand-robot swarm. *Science* **2014**, *345*, 795–799. [\[CrossRef\]](#) [\[PubMed\]](#)
- Yang, H.; Cao, S.; Bai, L.; Zhang, Z.; Kong, J. A distributed and parallel self-assembly approach for swarm robotics. *Robot. Auton. Syst.* **2019**, *118*, 80–92. [\[CrossRef\]](#)

27. Yang, H.; Kong, J.; Cao, S.; Duan, X.; Zhang, S. A distributed self-assembly approach for hollow shape in swarm robotics. *Int. J. Adv. Manuf. Technol.* **2020**, *108*, 2213–2230. [\[CrossRef\]](#)
28. Divband Soorati, M.; Zahadat, P.; Ghofrani, J.; Hamann, H. Adaptive path formation in self-assembling robot swarms by tree-like vascular morphogenesis. In *Distributed Autonomous Robotic Systems, Proceedings of the 14th International Symposium on Distributed Autonomous Robotic Systems (DARS 2018), Boulder, CO, USA, 15–17 October 2018*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 299–311.
29. Zhu, X.; Zhao, J.; Ji, J.; Fang, W.; Pan, Y. Shape Self-Assembly for a Swarm of Agents. In Proceedings of the 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC), Banff, AB, Canada, 9–11 May 2019; pp. 247–251.
30. Zheng, T.; Liu, Z.; Lin, H. Complex pattern generation for swarm robotic systems using spatial-temporal logic and density feedback control. In Proceedings of the 2020 American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 5301–5306.
31. Deshmukh, V.; Elamvazhuthi, K.; Biswal, S.; Kakish, Z.; Berman, S. Mean-field stabilization of Markov chain models for robotic swarms: Computational approaches and experimental results. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1985–1992. [\[CrossRef\]](#)
32. Klavins, E. Programmable self-assembly. *IEEE Contr. Syst. Mag.* **2007**, *27*, 43–56.
33. Mong-Ying, A.H.; Kumar, V. Pattern generation with multiple robots. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, 2006, ICRA 2006, Orlando, FL, USA, 15–19 May 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 2442–2447.
34. Groß, R.; Dorigo, M. Evolution of solitary and group transport behaviors for autonomous robots capable of self-assembling. *Adapt. Behav.* **2008**, *16*, 285–305. [\[CrossRef\]](#)
35. Sperati, V.; Trianni, V.; Nolfi, S. Self-organised path formation in a swarm of robots. *Swarm Intell.* **2011**, *5*, 97–119. [\[CrossRef\]](#)
36. Khaldi, B.; Harrou, F.; Cherif, F.; Sun, Y. Toward emerging cubic-spline patterns with a mobile robotics swarm system. *IEEE Trans. Cogn. Dev. Syst.* **2021**, *14*, 565–577. [\[CrossRef\]](#)
37. Cheah, C.C.; Hou, S.P.; Slotine, J.J.E. Region-based shape control for a swarm of robots. *Automatica* **2009**, *45*, 2406–2411. [\[CrossRef\]](#)
38. Pereira, A.R.; Hsu, L. Adaptive formation control using artificial potentials for Euler-Lagrange agents. *IFAC Proceedings Volumes*. **2008**, *41*, 10788–10793. [\[CrossRef\]](#)
39. Mullin, J.W. *Crystallization*; Elsevier: Amsterdam, The Netherlands, 2001.
40. McDonald, M.A.; Salami, H.; Harris, P.R.; Lagerman, C.E.; Yang, X.; Bommaris, A.S.; Grover, M.A.; Rousseau, R.W. Reactive crystallization: A review. *React. Chem. Eng.* **2021**, *6*, 364–400. [\[CrossRef\]](#)
41. Beckmann, W. *Crystallization: Basic Concepts and Industrial Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
42. Lerman, K.; Martinoli, A.; Galstyan, A. A review of probabilistic macroscopic models for swarm robotic systems. In Proceedings of the Swarm Robotics: SAB 2004 International Workshop, Santa Monica, CA, USA, 17 July 2004; Revised Selected Papers 1. Springer: Berlin/Heidelberg, Germany, 2005; pp. 143–152.
43. Correll, N.; Hamann, H. Probabilistic modeling of swarming systems. In *Springer Handbook of Computational Intelligence*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 1423–1432.
44. Agassounon, W.; Martinoli, A.; Easton, K. Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes. *Auton. Robot.* **2004**, *17*, 163–192. [\[CrossRef\]](#)
45. Konur, S.; Dixon, C.; Fisher, M. Analysing robot swarm behaviour via probabilistic model checking. *Robot. Auton. Syst.* **2012**, *60*, 199–213. [\[CrossRef\]](#)
46. Schmickl, T.; Hamann, H.; Wörn, H.; Crailsheim, K. Two different approaches to a macroscopic model of a bio-inspired robotic swarm. *Robot. Auton. Syst.* **2009**, *57*, 913–921. [\[CrossRef\]](#)
47. Hamann, H.; Wörn, H. An analytical and spatial model of foraging in a swarm of robots. In Proceedings of the Swarm Robotics: Second International Workshop, SAB 2006, Rome, Italy, 30 September–1 October 2006; Revised Selected Papers 2. Springer: Berlin/Heidelberg, Germany, 2007; pp. 43–55.
48. Hamann, H.; Wörn, H. A framework of space-time continuous models for algorithm design in swarm robotics. *Swarm Intell.* **2008**, *2*, 209–239. [\[CrossRef\]](#)
49. Prorok, A.; Correll, N.; Martinoli, A. Multi-level spatial modeling for stochastic distributed robotic systems. *Int. J. Robot. Res.* **2011**, *30*, 574–589. [\[CrossRef\]](#)
50. Haghighat, B.; Thandiackal, R.; Mordig, M.; Martinoli, A. Probabilistic modeling of programmable stochastic self-assembly of robotic modules. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 4656–4663.
51. Matthey, L.; Berman, S.; Kumar, V. Stochastic strategies for a swarm robotic assembly system. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 1953–1958.
52. Napp, N.; Burden, S.; Klavins, E. The statistical dynamics of programmed self-assembly. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, 2006, ICRA 2006, Orlando, FL, USA, 15–19 May 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 1469–1476.
53. Ganesan, V.; Chitre, M. On stochastic self-assembly of underwater robots. *IEEE Robot. Autom. Lett.* **2016**, *1*, 251–258. [\[CrossRef\]](#)
54. De Yoreo, J.J.; Vekilov, P.G. Principles of crystal nucleation and growth. *Rev. Mineral. Geochem.* **2003**, *54*, 57–93. [\[CrossRef\]](#)
55. Kondrashova, D.; Valiullin, R. Freezing and melting transitions under mesoscale confinement: Application of the Kossel–Stranski crystal-growth model. *J. Phys. Chem. C* **2015**, *119*, 4312–4323. [\[CrossRef\]](#)

-
56. O’Keeffe, M.; Hyde, B.G. *Crystal Structures*; Courier Dover Publications: Mineola, NY, USA, 2020.
 57. Dhanaraj, G.; Byrappa, K.; Prasad, V.; Dudley, M. *Springer Handbook of Crystal Growth*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 2.
 58. Efremov, M.A.; Kholod, I.I. Swarm Robotics Foraging Approaches. In Proceedings of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), St. Petersburg, Russia, 27–30 January 2020; pp. 299–304.
 59. Woodman, O.J. *An Introduction to Inertial Navigation*; University of Cambridge, Computer Laboratory: Cambridge, UK, 2007. [[CrossRef](#)]
 60. Shao, J. *Mathematical Statistics*; Springer Science & Business Media: New York, NY, USA, 2003.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.